



(2)

NRL Memorandum Report 6573

AD-A227 859

Estimation of The Time of Arrival
of Underwater Acoustic Signals
by Spline Functions
II: Algorithms for Noisy Data

Charles K. Chui

Center for Approximation Theory
Texas A&M University
College Station, TX 77843

for

Underwater Sound Reference Detachment
Naval Research Laboratory
P.O. Box 568337
Orlando, FL 32856-8337

DTIC
ELECTED
OCT 22 1990
S B D
C

1 October 1990

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

Public comment burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
	1 October 1990	Final Jun 88 til Apr 90	
4. TITLE AND SUBTITLE	5. FUNDING NUMBERS		
Estimation of The Time-of-Arrival of Underwater Acoustic Signals by Spline Functions - II: Algorithms for Noisy Data	PE #62314N		
6. AUTHOR(S)	Task #RJ14-121-0T0		
Charles K. Chui	WU Assn: DN880-326		
WU #59-0593-0-0			
Contracts: N62190-88-M-0409 & N62190-89-M-1536			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER		
Center for Approximation Theory Texas A&M University College Station, TX 77843			
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
Measurements Branch Underwater Sound Reference Detachment Naval Research Laboratory P.O. Box 568337, Orlando, FL 32856-8337	NRL Memorandum Report 6573		
11. SUPPLEMENTARY NOTES			
This work was supported by the Transducer Technology Project sponsored by the Office of Naval Technology (ONT 231). Part I of this work is reported in NRL Memorandum Report 6572 dated 1 Oct 90.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE		
Approved for public release; distribution unlimited.			
13. ABSTRACT (Maximum 200 words)			
The spline model in the previous report with the same title but a different subtitle (I: An Introduction) is modified by introducing a stacked initial knot, to represent the time-of-arrival, so that higher order splines with possibly non-zero derivatives at the initial knot can be used to represent acoustic signals with sharp increase in magnitude at the time-of-arrival. The main objective of this report is to provide two useful algorithms for determining the time-of-arrival using this modified (adaptive) spline model, when the acoustic signal to be measured is contaminated with noise. All relevant matrices in the penalized least-squares optimization model are analyzed so that the method of generalized cross-validation can be modified and extended for our study. Algorithms based on SVD and tridiagonalization with illustrative flow charts are given.			
14. SUBJECT TERMS	15. NUMBER OF PAGES		
Underwater acoustic signals; time-of-arrival; adaptive spline model; penalized least-squares; generalized cross-validation; singular valued decomposition; tridiagonalization; algorithms	53		
16. PRICE CODE			
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 128, Rev. 3-24
GSA FPMR (41 CFR) 101-11.4
101-11.4

(Blank Page)

CONTENTS

INTRODUCTION	1
SPLINE REPRESENTATIONS OF ACOUSTIC SIGNALS	1
<i>B</i> -Splines With Non-Zero Initial Derivatives	2
The Coefficient Matrices in L^2 Approximation	4
Derivation of the Coefficient Matrix $A_{3,h}$	6
Estimation by Cubic Splines and Algorithm I	9
PENALIZED LEAST-SQUARES ESTIMATION	12
Spline Solution of the Minimization Problem	14
Modification of the Extremal Problem	17
Computation of the Matrix $B_{k,h}^0$	19
The Generalized Cross-Validation Function	24
ALGORITHMS FOR NEAR-OPTIMAL ESTIMATION	27
Two Formulations of the Generalized Cross-Validation Function	28
Formulation of the Error Functional	31
An Algorithm for Determining the Signal Onset: Algorithm II	33
Another Algorithm for Determining the Signal Onset: Algorithm III	40
REFERENCES	49



Accession For	
NTIS	CP&I
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification _____	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or
	Special
A-1	

(Blank Page)

ESTIMATION OF TIME-OF-ARRIVAL OF UNDERWATER ACOUSTIC SIGNALS BY SPLINE FUNCTIONS

INTRODUCTION

An important problem in underwater signal processing research is the determination of the signal onset (or time-of-arrival) of an acoustic signal from possibly noisy discrete data. Once an accurate estimation of this value is obtained, the transducer and panel transfer function identification techniques can be applied. In the report [1], the underwater acoustic signal is represented by a spline curve with equally spaced knots such that the initial knot that lies in the interior of the time interval clearly indicates when the curve "takes off" to the right, and hence, can be used to define the signal onset. In particular, if the signal arrives fairly sharply a linear spline curve can be used to model the signal. However, if the signal arrives and increases in magnitude smoothly and slowly, a higher order spline curve such as a cubic spline is recommended. In a recent report [2], it was pointed out with several interesting examples that, indeed, a cubic spline curve does not give an accurate estimation of the signal onset when the signal function representation has nonzero first or second derivatives at the time-of-arrival. Since a cubic spline curve provides a better model than its linear counterpart to curve fitting, and hence, to representing an acoustic signal, we will modify the cubic spline model in Ref. 1 by using "stacked knots" at the initial knot position. As a consequence, the B -spline series is increased by two terms which may both drop out when the arriving signal has zero first and second derivatives at the signal onset, so that the approximation procedure is adaptive in nature.

The main objective of this report is to provide useful algorithms for determining the signal onset from discrete measurement of the acoustic signal which is contaminated with noise. The method of penalized least-squares discussed in the report [1] will be studied in great details. In particular, all relevant matrices for the setting of this particular problem will be analyzed so that the method of generalized cross-validation (GCV), (See [3-5]), can be modified and extended for our study. A global search procedure will then be used to determine a good estimation of the signal onset. For the sake of comparisons, algorithms based on SVD (singular valued decomposition) and tridiagonization will both be proposed. In addition, whenever it is appropriate, flow charts will be given to illustrate the algorithms.

SPLINE REPRESENTATIONS OF ACOUSTIC SIGNALS

Let $[0, d]$ denote the time interval on which an underwater acoustic signal is measured, and let $c = t_0, 0 \leq c < d$, represent the signal onset (or time-of-arrival) which is to be determined. Then the acoustic signal is represented by a spline curve with initial knot at t_0 . The knot sequence of this spline curve was given on page 7 of Ref. 1 by

$$t_h: \quad t_0 < t_1 < \cdots < t_n = d < t_{n+1} < \cdots < t_{n+k}$$

where $0 \leq t_0 = c < d$. Note that k knots t_{n+1}, \dots, t_{n+k} are tacked on to the right of the measurement interval $[0, d]$ to allow us to represent the last k B -splines in the

B-spline series representation of the spline curve with degree k . Recall that a *B*-spline with degree k is supported on $k + 2$ knots, so that the last *B*-spline has support given by $[t_{n-1}, t_{n+k}]$ which overlaps with the time interval $[0, d]$. As suggested by George and Muise [2], it is quite possible that the arrival of an underwater acoustic signal is not at all smooth in the sense that some of the initial derivative values, $f'(t_0), \dots, f^{(n-1)}(t_0)$, of the function $f(t)$ which represents the signal do not vanish. Hence, in order to use a k -th degree spline curve to represent the acoustic signal, we recommend in this report to extend the *B*-spline series to the left by stacking $k - 1$ knots at the initial knot so that, on one hand, non-zero initial derivative values $f^{(j)}(t_0), 1 \leq j \leq k - 1$, are allowed, and on the other hand, the spline curve still takes off at the initial knot t_0 . That is, we now consider the knot sequence

$$\mathbf{t}_h: t_{-k+1} = \dots = t_{-1} = t_0 < t_1 < \dots < t_n < \dots < t_{n+k} \quad (1)$$

where, again, $0 \leq t_0 = c < d$ and $t_n = d$. For the distinct knots t_0, \dots, t_{n+k} , we will still assume that they are equally spaced as in Ref. 1; that is, for $j = 1, \dots, n+k$, we set

$$t_j = t_{j-1} + h, \quad h = \frac{d - c}{n},$$

since there is nothing to gain by making the problem more complicated.

B-Splines With Non-Zero Initial Derivatives

To construct the *B*-splines with the new knot sequence \mathbf{t}_h , we return to Eq. (9) on page 5 of Ref. 1 by setting

$$B_{k,j}(t) = B_{k,\mathbf{t}_h,j}(t) = N_k\left(\frac{1}{h}(t - c) - j\right) \quad (2)$$

where $j = 0, \dots, n - 1$ and $N_k(t)$ denotes the *B*-spline with degree k and knots at the integers having support given by the interval $[0, k + 1]$. (See Fig. 1 below and on page 5 of Ref. 1).

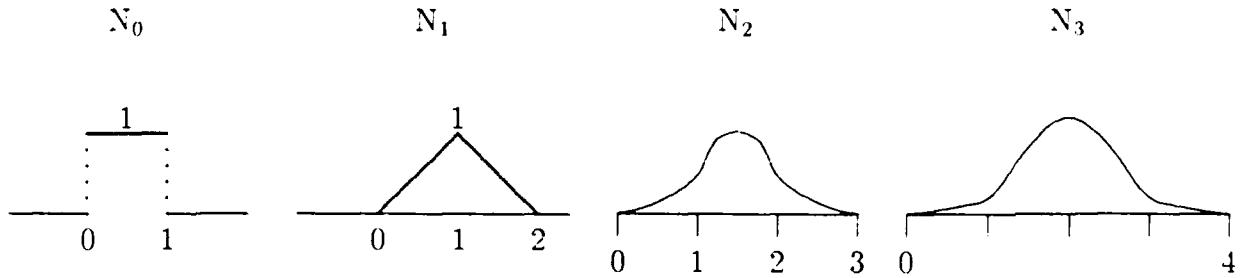


Fig. 1 - *B*-Splines of orders 1 - 4

For the knots t_{-k+1}, \dots, t_k which are no longer equally spaced, we must compute the *B*-splines $B_{k,j}(t)$, $j = -k + 1, \dots, -1$, without relying on $N_k(t)$. For this purpose, the method described on page 12 of Ref. 6 works quite well, since Bernstein coefficients must be obtained for later purposes. In this report, we will only consider linear and cubic splines (i.e., $k = 1$ and 3, respectively). Since no stacked knots are necessary for

linear splines, only cubic B -splines $B_{3,j}(t), j = -2, -1$, will be given. [For convenience, we will use the notation $B_j(t) = B_{3,j}(t)$.]

Using Bernstein representations as in Ref. 1, we have (for $k = 3$):

$$B_{-2}(t) = B_{3,t_h,-2}(t) = \tilde{N}_3\left(\frac{1}{h}(t - c)\right) \quad (3)$$

and

$$B_{-1}(t) = B_{3,t_h,-1}(t) = \hat{N}_3\left(\frac{1}{h}(t - c)\right), \quad (4)$$

where the Bernstein coefficients of \tilde{N}_3 and \hat{N}_3 are given in Fig. 2.

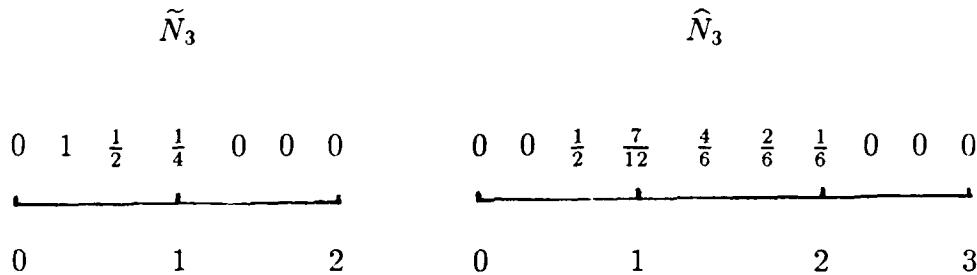


Fig. 2 - Bernstein coefficients of \tilde{N}_3 and \hat{N}_3

Of course, as mentioned in Eq. (2), for $j = 0, 1, \dots, n - 1$, we have

$$B_j(t) = B_{3,t_h,j}(t) = N_3\left(\frac{1}{h}(t - c) - j\right), \quad (5)$$

and the Bernstein representation of $N_3(t)$ was already computed on page 6 of Ref. 1 as shown in Fig. 3 below.

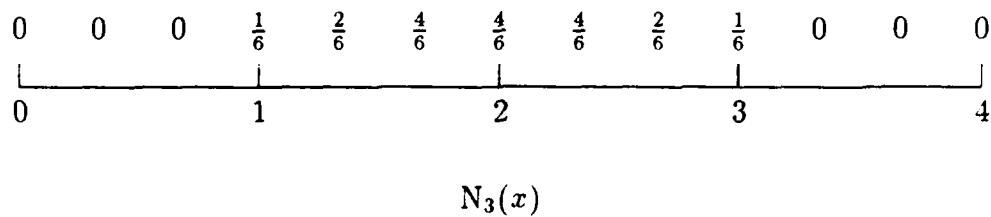


Fig. 3 - Bernstein representation of $N_3(t)$

In the usual Cartesian formulation, we may also write:

$$\tilde{N}_3(t) = \begin{cases} 3(1-t)^2t + \frac{3}{2}(1-t)t^2 + \frac{1}{4}t^3 & \text{if } 0 \leq t \leq 1 \\ \frac{1}{4}(2-t)^3 & \text{if } 1 \leq t \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

$$\widehat{N}_3(t) = \begin{cases} \frac{3}{2}(1-t)t^2 + \frac{7}{12}t^3 & \text{if } 0 \leq t \leq 1 \\ \frac{7}{12}(2-t)^3 + 2(2-t)^2(t-1) + (2-t)(t-1)^2 + \frac{1}{6}(t-1)^3 & \text{if } 1 \leq t \leq 2 \\ \frac{1}{6}(3-t)^3 & \text{if } 2 \leq t \leq 3 \\ 0 & \text{otherwise} \end{cases}$$

and

$$N_3(t) = \begin{cases} \frac{1}{6}t^3 & \text{if } 0 \leq t \leq 1 \\ \frac{1}{6}(2-t)^3 + (2-t)^2(t-1) + 2(2-t)(t-1)^2 + \frac{2}{3}(t-1)^3 & \text{if } 1 \leq t \leq 2 \\ \frac{2}{3}(3-t)^3 + 2(3-t)^2(t-2) + (3-t)(t-2)^2 + \frac{1}{6}(t-2)^3 & \text{if } 2 \leq t \leq 3 \\ \frac{1}{6}(4-t)^3 & \text{if } 3 \leq t \leq 4 \\ 0 & \text{otherwise.} \end{cases}$$

Using the notation in Eqs. (3) to (5), any cubic spline function $S_3(t)$ that represents the underwater acoustic signal $f(t)$ with signal onset at $t_0 = c$ and possibly nonzero values of $f'(t_0)$ and $f''(t_0)$ is given by the cubic spline series

$$S_3(t) = \sum_{j=-2}^{n-1} c_j B_j(t) \quad (6)$$

with knot sequence

$$\mathbf{t}_h : t_{-2} = t_{-1} = t_0 < \cdots < t_n < \cdots < t_{n+3} \quad (7)$$

where $0 \leq t_0 = c < d$, $t_n = d$, and $h = (d-c)/n$. Observe that Eq. (6) differs from Eq. (11) on page 7 of Ref. 1 in that two extra terms have been introduced to allow an adaptive curve fitting scheme by this new cubic spline representation.

The Coefficient Matrices in L^2 Approximation

For $L^2 = L^2[0, d]$ least-squares estimation of underwater acoustic signals, it is essential to compute the coefficient matrices

$$A_{k,h} = [b_{ij}^k]$$

where $-k+1 \leq i, j \leq n-1$ and

$$b_{ij}^k = \int_0^d B_{k,\mathbf{t}_h,j}(t) B_{k,\mathbf{t}_h,i}(t) dt. \quad (8)$$

Hence, for linear splines, where $k = 1$, the coefficient matrix

$$A_{1,h} = \frac{h}{6} \begin{bmatrix} 4 & 1 & & & \circ \\ 1 & 4 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 4 & 1 \\ \circ & & & 1 & 2 \end{bmatrix}, \quad (9)$$

given in Eq. (39) on page 15 in Ref. 1 is unchanged. For cubic splines, where $k = 3$, however, the dimension of the matrix $A_{3,h}$ is increased by 2, becoming an $(n+2) \times (n+2)$ matrix. In the next subsection, we will show that

$$A_{3,h} = \frac{h}{7!} \begin{bmatrix} \frac{1}{2}E_2 & \frac{1}{2}D_1 & \circ \\ \frac{1}{2}D_1^T & C_{n-3} & D \\ \circ & D^T & E_3 \end{bmatrix} \quad (10)$$

[compare with Eq. (40) on page 15 of Ref. 1], where

$$E_2 = \begin{bmatrix} 2232 & 1575 \\ 1575 & 3294 \end{bmatrix} \quad (11)$$

is a 2×2 block,

$$D_1 = \begin{bmatrix} 348 & 3 & 0 & 0 & \cdots & 0 \\ 2264 & 239 & 2 & 0 & \cdots & 0 \end{bmatrix} \quad (12)$$

is a $2 \times (n-3)$ block,

$$C_{n-3} = \begin{bmatrix} 2416 & 1191 & 120 & 1 & & & \circ \\ 1191 & 2416 & 1191 & & & & \\ 120 & 1191 & \ddots & \ddots & \ddots & & \\ 1 & & \ddots & \ddots & \ddots & \ddots & 1 \\ & \ddots & \ddots & \ddots & \ddots & \ddots & 120 \\ & & \ddots & \ddots & \ddots & \ddots & 1191 \\ \circ & & & & & & \\ & & & 1 & 120 & 1191 & 2416 \end{bmatrix} \quad (13)$$

is an $(n-3) \times (n-3)$ banded Toeplitz symmetric matrix as Eq. (41) on page 15 of Ref. 1.

$$D = \begin{bmatrix} 0 & \circ \\ 1 & 0 \\ 120 & 1 & 0 \\ 1191 & 120 & 1 \end{bmatrix} \quad (14)$$

is an $(n-3) \times 3$ Toeplitz matrix as in Eq. (42) on page 15 of Ref. 1, and

$$E_3 = \begin{bmatrix} 2396 & 1062 & 60 \\ 1062 & 1208 & 129 \\ 60 & 129 & 20 \end{bmatrix} \quad (15)$$

is a 3×3 block as in Eq. (43) on page 15 of Ref. 1. Of course, D_1^T denotes the transpose of D_1 in Eq. (12) and D^T the transpose of D in Eq. (14).

Derivation of the Coefficient Matrix $A_{3,h}$

To derive the coefficient matrix $A_{3,h}$ in Eq. (10), the Bernstein representations of $\tilde{N}_3(t)$, $\hat{N}_3(t)$, and $N_3(t)$ shown in Figs. 2 and 3 are used together with the relationships in Eqs. (3) to (5) in the integral in Eq. (8) for b_{ij}^3 (with $k = 3$). After a change of variables from $B_{3,t_h,i}(t)$ to $\tilde{N}_3(t)$, $\hat{N}_3(t)$, or $N_3(t)$, the integrals can be evaluated by using a formula in Eq. (37) on page 12 in Ref. 1, namely:

$$\int_r^{r+1} P_k Q_k = \frac{(k!)^2}{(2k+1)!} \sum_{\ell+m=k} \sum_{p+q=k} \frac{(\ell+p)!(m+q)!}{\ell!m!p!q!} c_{\ell m}^k d_{pq}^k. \quad (16)$$

where $k = 3$ is used for cubic splines. In the following, the superscript k for $c_{\ell m}^k$ and d_{pq}^k will be dropped for convenience.

(a) For $b_{-2,-2}^3$, we use $\tilde{N}_3(t)$ so that

$$\{c_{\ell m}\} = \left(0, 1, \frac{1}{2}, \frac{1}{4}\right), \quad \{d_{pq}\} = \left(\frac{1}{4}, 0, 0, 0\right). \quad (17)$$

yielding:

$$\begin{aligned} b_{-2,-2}^3 &= \frac{(3!)^2}{7!} \left[20 \cdot \left(\frac{1}{4}\right)^2 + 12 + 9 \cdot \left(\frac{1}{2}\right) + 4 \left(\frac{1}{4}\right) \right. \\ &\quad + 9 \cdot \left(\frac{1}{2}\right) + 12 \cdot \left(\frac{1}{2}\right)^2 + 10 \cdot \left(\frac{1}{2}\right) \cdot \left(\frac{1}{4}\right) \\ &\quad \left. + 4 \cdot \left(\frac{1}{4}\right) + 10 \cdot \left(\frac{1}{2}\right) \cdot \left(\frac{1}{4}\right) + 20 \cdot \left(\frac{1}{4}\right)^2 \right] h \\ &= \frac{h}{2 \cdot 7!} \cdot 2232. \end{aligned}$$

(b) For $b_{-2,-1}^3 = b_{-1,-2}^3$, we use both $\tilde{N}_3(t)$ and $\hat{N}_3(t)$ so that in addition to Eq. (17), we also need

$$\{c_{\ell m}\} = \left(0, 0, \frac{1}{2}, \frac{7}{12}\right), \quad \{d_{pq}\} = \left(\frac{7}{12}, \frac{4}{6}, \frac{2}{6}, \frac{1}{6}\right). \quad (18)$$

yielding:

$$\begin{aligned}
 b_{-2,-1}^3 &= b_{-1,-2}^3 = \frac{(3!)^2}{7!} [20(c_{03}\bar{c}_{03} + d_{03}\bar{d}_{03}) + 10(c_{12}\bar{c}_{03} + d_{12}\bar{d}_{03}) \\
 &\quad + 4(c_{21}\bar{c}_{03} + d_{21}\bar{d}_{03}) + (c_{30}\bar{c}_{03} + d_{30}\bar{d}_{03}) + 10(c_{03}\bar{c}_{12} + d_{03}\bar{d}_{12}) \\
 &\quad + 12(c_{12}\bar{c}_{12} + d_{12}\bar{d}_{12}) + 9(c_{21}\bar{c}_{12} + d_{21}\bar{d}_{12}) + 4(c_{30}\bar{c}_{12} + d_{30}\bar{d}_{12}) \\
 &\quad + 4(c_{03}\bar{c}_{21} + d_{03}\bar{d}_{21}) + 9(c_{12}\bar{c}_{21} + d_{12}\bar{d}_{21}) + 12(c_{21}\bar{c}_{21} + d_{21}\bar{d}_{21}) \\
 &\quad + 10(c_{30}\bar{c}_{21} + d_{30}\bar{d}_{21}) + (c_{03}\bar{c}_{30} + d_{03}\bar{d}_{30}) + 4(c_{12}\bar{c}_{30} + d_{12}\bar{d}_{30}) \\
 &\quad + 10(c_{21}\bar{c}_{30} + d_{21}\bar{d}_{30}) + 20(c_{30}\bar{c}_{30} + d_{30}\bar{d}_{30})]h \\
 &= \frac{h}{2 \cdot 7!} \cdot 1575.
 \end{aligned}$$

(c) For $b_{-1,-1}^3$, we also need the Bernstein coefficients

$$\{\bar{e}_{\ell m}\} = \left(\frac{1}{6}, 0, 0, 0\right) \quad (19)$$

in addition to Eq. (18) for $\hat{N}_3(t)$, yielding:

$$\begin{aligned}
 b_{-1,-1}^3 &= \frac{(3!)^2}{7!} \left[20\left(\left(\frac{7}{12}\right)^2 + \left(\frac{1}{6}\right)^2\right) + 10\left(\frac{7}{12} \cdot \frac{4}{6}\right) + 4\left(\frac{2}{6} \cdot \frac{7}{12}\right) \right. \\
 &\quad + \left(\frac{1}{6} \cdot \frac{7}{12}\right) + 10\left(\frac{7}{12} \cdot \frac{4}{6}\right) + 12\left(\frac{4}{6}\right)^2 + 9\left(\frac{2}{6} \cdot \frac{4}{6}\right) \\
 &\quad + 4\left(\frac{1}{6} \cdot \frac{4}{6}\right) + 4\left(\frac{7}{12} \cdot \frac{2}{6}\right) + 9 \cdot \left(\frac{4}{6} \cdot \frac{2}{6}\right) + 12\left(\left(\frac{1}{2}\right)^2 + \left(\frac{2}{6}\right)^2\right) \\
 &\quad + 10\left(\frac{1}{2} \cdot \frac{7}{12} + \frac{1}{6} \cdot \frac{2}{6}\right) + \left(\frac{7}{12} \cdot \frac{1}{6}\right) + 4\left(\frac{4}{6}\right)\left(\frac{1}{6}\right) \\
 &\quad \left. + 10\left(\frac{1}{2} \cdot \frac{7}{12} + \frac{2}{6} \cdot \frac{1}{6}\right) + 20\left(\left(\frac{7}{12}\right)^2 + \left(\frac{1}{6}\right)^2\right) \right]h \\
 &= \frac{h}{2 \cdot 7!} \cdot 3294.
 \end{aligned}$$

Combining the results in (a)-(c), we have obtained the blocks E_2 and E_2^T in Eq. (11). We will next verify the correctness of the blocks D_1 and D_1^T .

(d) For $b_{-2,0}^3 = b_{0,-2}^3$, we need the Bernstein coefficients

$$\{\hat{c}_{\ell m}\} = \left(0, 0, 0, \frac{1}{6}\right), \quad \{\hat{d}_{pq}\} = \left(\frac{1}{6}, \frac{2}{6}, \frac{4}{6}, \frac{4}{6}\right) \quad (20)$$

of $N_3(t)$ in addition to those in Eq. (17) of $\tilde{N}_3(t)$, yielding:

$$\begin{aligned}
 b_{-2,0}^3 &= b_{0,-2}^3 = \frac{36}{7!} \left[20\left(\frac{4}{6}\right)\left(\frac{1}{6}\right) + 10\left(\frac{1}{4}\right)\left(\frac{2}{6}\right) + 4\left(\frac{1}{6}\right)\left(\frac{4}{6}\right) \right. \\
 &\quad + \left(\frac{1}{4}\right)\left(\frac{4}{6}\right) + 4\left(\frac{1}{6}\right) + 10\left(\frac{1}{2} \cdot \frac{1}{6}\right) + 20\left(\frac{1}{4} \cdot \frac{1}{6}\right) \left. \right]h \\
 &= \frac{h}{2 \cdot 7!} \cdot 348.
 \end{aligned}$$

(e) For $b_{-2,1}^3 = b_{1,-2}^3$, we only need the Bernstein coefficients $\{\hat{c}_{\ell m}\}$ of $N_3(t)$ in Eq. (20) as well as those of $\tilde{N}_3(t)$ in Eq. (17), yielding:

$$b_{-2,1}^3 = b_{1,-2}^3 = \frac{36}{7!} \left[\frac{1}{4} \cdot \frac{1}{6} \right] h = \frac{h}{2 \cdot 7!} \cdot 3.$$

(f) Since the supports of $\tilde{N}_3(t)$ and $N_3(t-j)$, for $j \geq 2$, do not overlap, we have $b_{-2,j}^3 = b_{j,-2}^3 = 0$ for all $j \geq 2$.

(g) For $b_{-1,0}^3 = b_{0,-1}^3$, we need the Bernstein coefficients of $\tilde{N}_3(t)$ in Eqs. (18) and (19), and those of $N_3(t)$ in Eq. (20) and

$$\{\hat{e}_{pq}\} = \left(\frac{4}{6}, \frac{4}{6}, \frac{2}{6}, \frac{1}{6} \right) \quad (21)$$

yielding:

$$\begin{aligned} b_{-1,0}^3 = b_{0,-1}^3 &= \frac{(3!)^2}{7!} \sum_{\ell=0}^3 \sum_{p=0}^3 \frac{(\ell+p)!(6-\ell-p)!}{\ell!(3-\ell)!p!(3-p)!} [\hat{c}_{\ell}\hat{c}_p + \hat{d}_{\ell}\hat{d}_p + \hat{e}_{\ell}\hat{e}_p] h \\ &= \frac{h}{2 \cdot 7!} (2264), \end{aligned}$$

where the second subscripts m and q in $\hat{c}_{\ell m}, \hat{e}_{pq}$, etc. have been deleted for convenience.

(h) For $b_{-1,1}^3 = b_{1,-1}^3$, we need $\{\bar{d}_{pq}\}$ in Eq. (18), $\{\hat{e}_{\ell m}\}$ in Eq. (19), and $\{\hat{c}_{\ell m}\}, \{\hat{d}_{pq}\}$ in Eq. (20), yielding

$$\begin{aligned} b_{-1,1}^3 = b_{1,-1}^3 &= \frac{36}{7!} \left[20 \left(\frac{1}{6} \cdot \frac{1}{6} \right) + 10 \left(\frac{1}{6} \cdot \frac{2}{6} \right) \right. \\ &\quad + 4 \left(\frac{1}{6} \cdot \frac{4}{6} \right) + \left(\frac{7}{12} \cdot \frac{1}{6} + \frac{1}{6} \cdot \frac{4}{6} \right) + 4 \left(\frac{4}{6} \cdot \frac{1}{6} \right) \\ &\quad \left. + 10 \left(\frac{2}{6} \cdot \frac{1}{6} \right) + 20 \left(\frac{1}{6} \cdot \frac{1}{6} \right) \right] h \\ &= \frac{h}{2 \cdot 7!} \cdot 239. \end{aligned}$$

(i) For $b_{-1,2}^3 = b_{2,-1}^3$, we only need $\{\bar{e}_{pq}\}$ and $\{\hat{c}_{\ell m}\}$, yielding:

$$b_{-1,2}^3 = b_{2,-1}^3 = \frac{36}{7!} \left[\frac{1}{6} \cdot \frac{1}{6} \right] h = \frac{h}{2 \cdot 7!} \cdot 2.$$

(j) For $j \geq 3$, we have $b_{-1,j}^3 = b_{j,-1}^3 = 0$, since the supports of the two B-splines $\tilde{N}_3(t)$ and $N_3(t-j)$, where $j \geq 3$, do not overlap.

Combining the results in (d)-(j), we have obtained the blocks D_1 and D_1^T in Eq. (12) as well as the two zero blocks of $A_{3,h}$ at the upper-right and lower-left corners in Eq. (10). Since the other blocks C_{n-3}, D, D^T , and E_3 of $A_{3,h}$ have been verified in Ref. 1, we have now obtained the formulation of Eq. (10) for $A_{3,h}$.

Estimation by Cubic Splines

For noise-free or very low noise signals, we may use $L^2 = L^2[0, d]$ approximation as suggested in Ref. 1. Let $(\tau_i, f_i), i = 1, \dots, N$, and $0 \leq \tau_1 < \dots < \tau_N \leq d$, denote the data measurement, where the number N of sample points is usually much larger than the number n of knots of the cubic B -spline series Eq. (6). As in Ref. 1, let $\hat{f}(t)$ denote the piecewise linear continuous function on $[0, d]$, linear on each interval $[\tau_i, \tau_{i+1}]$ such that $\hat{f}(\tau_i) = f_i$. This continuous model of the discrete signal $\{\tau_i, f_i\}$ is recommended only in a noise-free or low-noise environment. As in Ref. 1 (pages 12-13), we consider the $L^2 = L^2[0, d]$ estimation of $\hat{f}(t)$ by a cubic spline curve given by the spline series $S_3(t)$ in Eq. (6). Fix the number n of knots ($n \ll N$), and for the time being also let $h (= (d - c)/n)$ be fixed. The L^2 model with “smoothing parameter” λ considered in Ref. 1 is to minimize the functional

$$K_3(h, \mathbf{c}) = \int_0^d \left| \hat{f}(t) - \sum_{j=-2}^{n-1} c_j B_j(t) \right|^2 dt + \lambda \left(\sum_{j=-2}^{n-1} c_j^2 \right) \quad (22)$$

over all c_{-2}, \dots, c_{n-1} , where

$$\mathbf{c} = \begin{bmatrix} c_{-2} \\ c_{-1} \\ \vdots \\ c_{n-1} \end{bmatrix}.$$

We will discuss this model in details in the section entitled Penalized Least-Squares Estimation. Here, the B -splines $B_j(t)$ are given by Eqs. (3) to (5). Let

$$\hat{S}_3(t) = \hat{S}_3(t; \lambda, h) = \sum_{j=-2}^{n-1} \hat{c}_j B_j(t) \quad (23)$$

be the (unique) minimum solution; that is,

$$E_3(h) := K_3(h, \hat{\mathbf{c}}) = \min_{\mathbf{c}} K_3(h, \mathbf{c}) \quad (24)$$

where $\hat{\mathbf{c}} = [\hat{c}_{-2}, \dots, \hat{c}_{n-1}]^T$ depends on λ and h . Then $\hat{\mathbf{c}} = \hat{\mathbf{c}}(\lambda, h)$ can be determined by solving the normal equations

$$\sum_{j=-2}^{n-1} b_{ij}^2 \hat{c}_j + \lambda \hat{c}_i = f_{h,i}^o, \quad i = -2, \dots, n-1 \quad (25)$$

where

$$f_{h,i}^o = \int_0^d \hat{f}(t) B_i(t) dt. \quad (26)$$

In matrix form, Eq. (24) becomes

$$(A_{3,h} + \lambda I_{n+2})\hat{\mathbf{c}} = \mathbf{f}_h^\circ \quad (27)$$

where $A_{3,h}$ is the coefficient matrix given by Eq. (10), I_{n+2} is the $(n+2) \times (n+2)$ identity matrix, and

$$\mathbf{f}_h^\circ = \begin{bmatrix} f_{h,-2}^\circ \\ \vdots \\ f_{h,n-1}^\circ \end{bmatrix}$$

is the $(n+2)$ -dimensional data vector. To compute \mathbf{f}_h° , a change of variables in Eq. (25) yields:

$$f_{h,-2}^\circ = h \int_0^2 \hat{f}(h(t-n) + d) \tilde{N}(t) dt, \quad (28)$$

$$f_{h,-1}^\circ = h \int_0^3 \hat{f}(h(t-n) + d) \tilde{N}(t) dt, \quad (29)$$

and for $i = 0, \dots, n-1$,

$$f_{h,i}^\circ = h \int_0^{n-i} \hat{f}(h(t-n+i) + d) N(t) dt. \quad (30)$$

Note that in Eq. (30) the integral is taken over the interval $[0, 4]$ for $i = 0, \dots, n-4$, but is taken over a smaller interval for $i = n-3, n-2, n-1$. According to page 32 and Theorem 4 on page 25 in Ref. 1, to determine the signal onset t_0 , we must determine the set $H = \{\tilde{h}\}$ such that each \tilde{h} satisfies:

$$E_3(\tilde{h}) = \min_{h \geq 0} E_3(h) \quad (31)$$

where $E_3(h) = K_3(h, \hat{c}(\lambda, h))$ is defined in Eq. (24). Here, since only noise-free or low-noise environment is considered, λ is fixed and in fact may be set to be zero for noise-free signals or a very small positive number for signals with low-noise contamination. Then the signal onset (or time-of-arrival) is given by

$$t_0 = d - nh^* \quad (32)$$

where h^* is the smallest number in $H = \{\tilde{h}\}$.

In Ref. 1 (pages 25-26), to determine the set $H = \{\tilde{h}\}$, we set $B = (1/h)A_{3,h}$ and $\mathbf{b} = (1/h)\mathbf{f}^\circ$; i.e.,

$$\begin{cases} B = \left[\frac{1}{h} b_{ij}^3 \right]_{-2 \leq i,j \leq n-1} \\ \mathbf{b} = \left[\frac{1}{h} f_i^\circ \right]_{-2 \leq i \leq n-1} \end{cases} \quad (33)$$

[which is essentially Eq. (74) in Ref. 1], and write

$$B = P \Lambda P^T \quad (34)$$

where

$$\Lambda = \begin{bmatrix} \lambda_1 & & \circ \\ & \ddots & \\ \circ & & \lambda_{n+2} \end{bmatrix} \quad (35)$$

with $\lambda_1 \geq \dots \geq \lambda_{n+2} > 0$ being the eigenvalues of B and

$$P = [\mathbf{u}_1 \dots \mathbf{u}_{n+2}] \quad (36)$$

with \mathbf{u}_i being the eigenvector of B corresponding to λ_i and having unit length. Also, set

$$\Gamma = \begin{bmatrix} \frac{h}{h\lambda_1 + \lambda} & & \circ \\ & \ddots & \\ \circ & & \frac{h}{h\lambda_{n+2} + \lambda} \end{bmatrix} \quad (37)$$

and

$$\hat{\mathbf{b}} = P^T \mathbf{b} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_{n+2} \end{bmatrix} \quad (38)$$

where \mathbf{b} is defined in Eq. (33). Then $E_3(h)$ in Eq. (24) becomes

$$E_3(h) = \int_0^d |\hat{f}(t)|^2 dt - \sum_{i=1}^{n+2} \frac{h^2}{h\lambda_i + \lambda} \hat{b}_i^2 \quad (39)$$

where \hat{b}_i is the i -th component of $\hat{\mathbf{b}}$ in Eq. (38). Hence, each \tilde{h} is obtained by finding an *absolute maximum* of the expression

$$T_3(h) = \sum_{i=1}^{n+2} \frac{d_i^2}{h\lambda_i + \lambda} \quad (40)$$

where

$$d_i = d_i(h) = h\hat{b}_i. \quad (41)$$

We summarize the above discussion as follows.

Algorithm I (Cubic Spline Estimation of Signal Onset in Noise-Free or Low-Noise Situations)

- (1°) Choose $\lambda = 0$ (if the signal is noise-free) or a positive but small value of λ (for low-noise signal). (In the next two sections, we will discuss procedures for estimating λ when the signal is fairly noisy.) Also, choose a positive integer n such as $n = 10$ or anything larger. (The dimension of the matrix will be $(n+2)$ by $(n+2)$.)
- (2°) Compute the eigenvalue-eigenvector pairs $(\lambda_i, \mathbf{u}_i)$, $i = 1, \dots, n+2$, of the matrix B in Eq. (33) using the entries b_{ij}^3 of A_{3h} given by Eqs. (10) to (15), where \mathbf{u}_i is normalized to have unit length. Note that the variable h does not appear at this stage.

(3°) Form the matrices P and Γ .

(4°) Compute the data vector $\mathbf{f}^o = [f_{-2}^o, \dots, f_{n-1}^o]^T$ in Eqs. (28) to (30) and

$$\mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_{n+2} \end{bmatrix} = P^T \begin{bmatrix} f_{-2}^0 \\ \vdots \\ f_{n-1}^0 \end{bmatrix}. \quad (43)$$

(5°) Plot the curve for $T_3(h)$ in Eq. (40) and determine its set of absolute maxima $H = \{\hat{h}\}$.

(6°) Determine the smallest value h^* in H . Then the signal onset is $t_0 = d - nh^*$.

(7°) Compute

$$\hat{\mathbf{c}} = P\Gamma\hat{\mathbf{b}} \quad (44)$$

where $\hat{\mathbf{c}} = [\hat{c}_{-2}, \dots, \hat{c}_{n-1}]^T$.

(8°) Plot the cubic spline curve

$$\hat{S}_3(t) = \sum_{i=-2}^{n-1} \hat{c}_i B_i(t) \quad (45)$$

where

$$\begin{aligned} B_{-2}(t) &= \tilde{N}_3\left(\frac{1}{h^*}(t - t_0)\right) \\ B_{-1}(t) &= \hat{N}_3\left(\frac{1}{h^*}(t - t_0)\right) \end{aligned}$$

and

$$B_j(t) = N_3\left(\frac{1}{h^*}(t - t_0) - j\right)$$

for $j = 0, \dots, n-2$. [See Eqs. (3) to (5)]. Then $\hat{S}_3(t)$ is the cubic spline estimate of the underwater acoustic signal with signal onset at t_0 .

PENALIZED LEAST-SQUARES ESTIMATION

We are interested in fitting a set of noisy data $\{\tau_i, f_i\}$, $i = 1, \dots, N$, where $0 \leq \tau_1 < \dots < \tau_N \leq d$, by a "smooth" curve $g(t)$ on $[c, d]$, which is continuous on $[0, d]$ such that $g(t) = 0$ for $0 \leq t \leq c$ but is *not* identically zero on $[0, c']$ for *any* $c' > c$. Hence, $c = t_0$ represents the signal onset of the underwater acoustic signal. We assume that the actual signal is given by a function $f^*(t)$ in the sense that

$$f_i = f^*(\tau_i) + \varepsilon_i \quad (46)$$

where ε_i 's are uncorrelated noise processes with zero mean and positive variances; that is,

$$\mathcal{E}(\varepsilon_i) = 0 \quad \text{and} \quad \mathcal{E}(\varepsilon_i^T \varepsilon_j) = \sigma_i^2 \delta_{ij}$$

where $\sigma_i > 0$, δ_{ij} is the Kronecker delta (which is defined to be 1 for $i = j$ and 0 for $i \neq j$), and \mathcal{E} is the expectation operator. Our choice of the “smoothing” data-fitting curve $g(t)$ is motivated by considering *nonparametric regressions* in the sense that no mathematical model of the smooth function $g(t)$ is a priori assumed. Here, “smoothness” means that $g(t)$ belongs to the class

$$W_{2,0}^m = \{g \in C^{m-1}[c, d]: g(c) = 0, g^{(m)}(t) \text{ square integrable}\} \quad (47)$$

of functions on $[c, d]$, having zero extension to the whole interval $[0, d]$, where $0 \leq c = t_0$. The subscripts 2 and 0 indicate that least-squares measurement will be used and the functions all vanish at $c = t_0$, respectively, and m is any preassigned non-negative integer of our choice of the order smoothness. Of course, $C^{m-1}[c, d]$ denotes, as usual, the collection of all functions $g(t)$ such that $g(t), g'(t), \dots, g^{(m-1)}(t)$ are all continuous on $[c, d]$. Note that $W_{2,0}^m$ is a proper subspace of the so-called Sobolev space W_2^m on $[c, d]$ where the condition $g(c) = 0$ is not assumed. It must be emphasized that any function $g(t)$ in $W_{2,0}^m$ is extended to $[0, d]$ by setting $g(t) = 0$ for $0 \leq t \leq c$ to represent the underwater acoustic signal. (We remark that our algorithms in this report can be modified to study parametric regressions using sinusoidal “wave” functions although the calculations would have to become much more complicated.)

Since the data (τ_i, f_i) are noisy, no reliable way is available to model the data by a continuous function $\hat{f}(t)$ as before, so that the L^2 norm cannot be used. Instead, we will use the $\ell^2(\mathbf{w})$ norm defined by

$$\|\mathbf{g}\|_2 = \|\mathbf{g}\|_{\ell^2(\mathbf{w})} = \left\{ \frac{1}{N} \sum_{i=1}^N (g_i)^2 w_i \right\}^{1/2} \quad (48)$$

where each w_i is strictly positive with $w_1 + \dots + w_N = N$ and $\mathbf{g} = \{g_i\}$. If $g_i = g(\tau_i)$ for some function $g(t)$, we will simply use the notation $\|g\|_2 = \|\mathbf{g}\|_2$. The sequence $\mathbf{w} = \{w_i\}$ is called the *weight* of the $\ell^2 = \ell^2(\mathbf{w})$ norm. When the variances σ_i^2 's are all non-zero as we have assumed here, the *optimal* weight is

$$w_i = \frac{N\sigma_i^{-2}}{\sigma_1^{-2} + \dots + \sigma_N^{-2}}, \quad (49)$$

the reciprocals of the variances so normalized that $w_1 + \dots + w_N = N$. (See [7], page 18). Hence, if the variances can be measured and each σ_i^2 is nonzero we will always choose w_i as in Eq. (49); but if they cannot be estimated, we will simply set $w_i = 1$ for all i . In the case when $N_0, 1 \leq N_0 \leq N$, of the σ_i^2 's are zero, then the weights corresponding to the zero σ_i^2 's are all equal to NN_0^{-1} , while the weight 0 is assigned to the remaining ones corresponding to nonzero σ_i^2 's. This weight assignment is simply a consequence of the limiting process from Eq. (49).

Given a data sequence $\{\tau_i, f_i\}$, we are interested in studying the minimization of the functional:

$$\begin{aligned} G_h(g, \lambda) &= \|g - \mathbf{f}\|_2^2 + \lambda \int_c^d (g^{(m)}(t))^2 dt \\ &= \frac{1}{N} \sum_{i=1}^N (g(\tau_i) - f_i)^2 w_i + \lambda \int_c^d (g^{(m)}(t))^2 dt \end{aligned} \quad (50)$$

where $\lambda \geq 0$ is called the *smoothing parameter*. Note that $\lambda = 0$ corresponds to strictly least-squares data fitting, but for very large values of λ , the term $\|g - f\|_2^2$ becomes negligible, so that the functional $G_h(g, \lambda)$ is dominated by the “smoothing” operator defined by the square of the m -th order derivative. Our objective is to find the *best* data fitting curve while guaranteeing certain degree of smoothness governed by m and λ ; that is, we are interested in obtaining a curve $g(t)$ that minimizes the functional $G_h(g, \lambda)$, or some modified formulations of it, over all functions $g(t)$ in $W_{2,0}^m$ and all smoothing parameters λ .

Since functions from $W_{2,0}^m$ are used, they have continuous extensions from $[c, d]$ to $[0, d]$ such that the extended values are identically zero. Hence, an “extremal function” in the above discussion represents the acoustic signal with signal onset at $c = t_0$. Consequently, determining the set $H = \{\tilde{h}\}$ such that each $\tilde{h} > 0$ satisfies

$$\min_{g \in W_{2,0}^m} G_{\tilde{h}}(g, \lambda^*) = \min_{h \geq 0} \min_{g \in W_{2,0}^m} G_h(g, \lambda^*), \quad (51)$$

where λ^* is characterized by minimization of the weighted mean-squares error in approximating the given data (or some modification of it) yields the signal onset $t_0 = d - nh^*$, where h^* is the minimum value among all \tilde{h} in H (See [1], pages 25-26. See subsection entitled The Generalized Cross-Validation Function in this report for a more precise notion of λ^*).

Spline Solution of the Minimization Problem

In the following we will motivate our approach to the study of our extremal problems using splines by studying the minimization problem:

$$\tilde{E}_m(h, \lambda) = \min_{g \in W_{2,0}^m} G_h(g, \lambda) \quad (52)$$

for any fixed positive values of h and λ . Here the data $\{\tau_i, f_i\}$ are to be estimated. We first remark that it is a standard mathematical argument (using the completeness of $W_{2,0}^m$) that a function $g^*(t)$ in $W_{2,0}^m$ exists, such that

$$G_h(g^*, \lambda) = \tilde{E}_m(h, \lambda). \quad (53)$$

Next, we will see that any extremal function $g^*(t)$ [satisfying Eq. (53)] must also satisfy the condition:

$$\int_c^d g^{*(m)}(t) \ell^{(m)}(t) dt = 0, \text{ all } \ell(t) \in \tilde{V}_{2,0}^m, \quad (54)$$

where $\tilde{V}_{2,0}^m$ is defined by

$$\tilde{V}_{2,0}^m = \{\ell \in W_{2,0}^m: \ell(\tau_i) = 0, \quad i = 1, \dots, N\}.$$

To verify this fact, let us consider the new data

$$z_i = g^*(\tau_i) \quad (55)$$

and the corresponding subcollection

$$\tilde{W}_{2,0}^m = \{g \in W_{2,0}^m: g(\tau_i) = z_i, \quad i = 1, \dots, N\}$$

of functions in $W_{2,0}^m$ that interpolate the same data as $g^*(t)$. Hence, we have

$$\begin{aligned} 0 &\leq G_h(g, \lambda) - G_h(g^*, \lambda) \\ &= \lambda \left\{ \int_c^d (g^{(m)}(t))^2 dt - \int_c^d (g^{*(m)}(t))^2 dt \right\} \end{aligned} \quad (56)$$

for all $g \in \widetilde{W}_{2,0}^m$. In particular, for any $\ell \in \widetilde{V}_{2,0}^m$ and any real number s , since $g^* + s\ell$ is in $\widetilde{W}_{2,0}^m$, the function

$$F(s) = \int_c^d (g^{*(m)}(t) + s\ell^{(m)}(t))^2 dt, \quad (57)$$

which is a real-valued differentiable function in s , has a relative minimum at $s = 0$, so that $F'(0) = 0$. In other words, we have verified Eq. (54).

Next, we will verify that $g^*(t)$ is essentially unique if the number \tilde{N} of τ_i 's that lie in the interval $[c, d]$ is at least m , in the sense that for any $g(t)$ in $\widetilde{W}_{2,0}^m$ that also satisfies Eq. (54) as $g^*(t)$ does, we have $g(t) \equiv g^*(t)$. Indeed, if $g(t)$ is in $\widetilde{W}_{2,0}^m$, then $g(t) - g^*(t)$ is in $\widetilde{V}_{2,0}^m$ and can be used as a function $\ell(t)$ in Eq. (54). Hence, since both $g(t)$ and $g^*(t)$ satisfy Eq. (54), their difference also satisfies Eq. (54), so that

$$\begin{aligned} 0 &= \int_c^d (g^{(m)}(t) - g^{*(m)}(t))\ell^{(m)}(t) dt \\ &= \int_c^d (g^{(m)}(t) - g^{*(m)}(t))^2 dt \end{aligned}$$

which implies that $g^{(m)}(t) - g^{*(m)}(t) \equiv 0$, or $g(t) - g^*(t)$ is a polynomial with degree at most m on $[c, d]$. But since this polynomial has \tilde{N} zeros and $\tilde{N} \geq m$, we may conclude that $g(t) - g^*(t) \equiv 0$.

The above observation allows us to characterize $g^*(t)$ as follows. Let $S(t)$ be a function in $C^{2m-2}[c, d]$ whose restriction to each of the intervals $[c, \tau_{N-\tilde{N}+1}]$, $[\tau_{N-\tilde{N}+1}, \tau_{N-\tilde{N}+2}]$, \dots , $[\tau_{N-1}, \tau_N]$, and $[\tau_N, d]$ is a polynomial with degree at most $2m-1$; i.e., $S(t)$ is a *spline* of degree $2m-1$ in $[c, d]$. Let us first assume that $c \neq \tau_{N-\tilde{N}+1}$ and $d \neq \tau_N$, so that there are precisely \tilde{N} interior knots: $\tau_{N-\tilde{N}+1}, \dots, \tau_N$ in $[c, d]$, and this implies that $S(t)$ has $2m + \tilde{N}$ free parameters, $2m$ from the polynomial piece on $[c, \tau_{N-\tilde{N}+1}]$, and 1 from each of the \tilde{N} interior knots in $[c, d]$. (The argument will be similar if one or both of $\tau_{N-\tilde{N}+1}$ and τ_N should become non-interior.) Hence, we may construct the (unique) spline $S(t)$ that satisfies the interpolatory conditions:

- (i) $S(c) = 0$,
- (ii) $S(\tau_i) = z_i$, for $i = N - \tilde{N} + 1, \dots, N$,
- (iii) $S^{(m)}(c) = \dots = S^{(2m-2)}(c) = 0$, and
- (iv) $S^{(m)}(d) = \dots = S^{(2m-1)}(d) = 0$;

a total of $1 + \tilde{N} + (m-1) + m = 2m + \tilde{N}$ conditions. If $\tau_{N-\tilde{N}+1} = c$, then the condition (i) is not necessary since it is included in (ii); and if $\tau_N = d$, then it will be seen later

that the condition $S^{(2m-1)}(d) = 0$ in (iv) can be deleted. In any case, the spline $S(t)$ satisfying (i)-(iv) (or the corresponding ones if $\tau_N = d$) exists and is actually unique. By setting $S(t) = 0$ for $0 \leq t \leq c$, it follows from condition (i) that $S(t)$ is in $W_{2,0}^m$. (Actually it is somewhat smoother, being in $C^{2m-2}[c, d]$, since $2m-2 \geq m-1$ for $m \geq 1$.) We will now verify that, in fact, $g^*(t) \equiv S(t)$ by showing that $S(t)$ satisfies Eq. (54) as $g^*(t)$ does. [Observe that $S(t)$ is in $\tilde{W}_{2,0}^m$ by the interpolatory condition (ii).] We remark that $S(t)$ is *not* a natural spline, since the condition $S^{(2m-1)}(c) = 0$ must be dropped and an interpolation condition $S(d) = z_{N+1}$ is required for it to qualify as a natural spline of order $2m$.

To verify Eq. (54), let $\ell(t)$ be any function in $\tilde{V}_{2,0}^m$. Then in addition to the interpolatory conditions $\ell(\tau_i) = 0$, we also have $\ell(c) = 0$. For notational convenience, we set $c = x_0, \tau_{N-\tilde{N}+1} = x_1, \dots, \tau_N = x_{\tilde{N}}$, and $d = x_{\tilde{N}+1}$, so that

$$\ell(x_i) = 0, \quad i = 0, \dots, \tilde{N} \quad (58)$$

and

$$\begin{cases} S(x_i) = z_i, & i = 0, \dots, \tilde{N}, \\ S^{(m)}(x_0) = \dots = S^{(2m-2)}(x_0) = 0, \\ S^{(m)}(x_{\tilde{N}+1}) = \dots = S^{(2m-1)}(x_{\tilde{N}+1}) = 0. \end{cases} \quad (59)$$

By applying integration by parts, we then have, for any $\ell(t)$ in $\tilde{V}_{2,0}^m$:

$$\begin{aligned} \int_c^d S^{(m)}(t) \ell^{(m)}(t) dt &= \sum_{i=1}^{\tilde{N}+1} \int_{x_{i-1}}^{x_i} S^{(m)}(t) \ell^{(m)}(t) dt \\ &= \sum_{i=1}^{\tilde{N}+1} \{ S^{(m)}(x_i) \ell^{(m-1)}(x_i) - S^{(m)}(x_{i-1}) \ell^{(m-1)}(x_{i-1}) \\ &\quad - \int_{x_{i-1}}^{x_i} S^{(m+1)}(t) \ell^{(m-1)}(t) dt \} = \dots \\ &= \sum_{i=1}^{\tilde{N}+1} \left\{ \sum_{j=0}^{m-1} (-1)^j [S^{(m+j)}(x_i) \ell^{(m-j-1)}(x_i) - S^{(m+j)}(x_{i-1}) \ell^{(m-j-1)}(x_{i-1})] \right. \\ &\quad \left. + (-1)^m \int_{x_{i-1}}^{x_i} S^{(2m)}(t) \ell(t) dt \right\} \\ &= \sum_{j=0}^{m-1} (-1)^j [S^{(m+j)}(x_{\tilde{N}+1}) \ell^{(m-j-1)}(x_{\tilde{N}+1}) - S^{(m+j)}(x_0) \ell^{(m-j-1)}(x_0)] \\ &\quad + (-1)^m \sum_{i=1}^{\tilde{N}+1} \int_{x_{i-1}}^{x_i} S^{(2m)}(t) \ell(t) dt = 0. \end{aligned}$$

by telescoping, applying Eqs. (58) and (59), and using the fact that $S^{(2m)}(t) = 0$ on each $[x_{i-1}, x_i]$, respectively. This completes the proof of the claim that $g^*(t) \equiv S(t)$.

Modification of the Extremal Problem

We have just proved that spline functions **with knots given by the sample points** $\{\tau_i\}$ where the data $\{f_i\}$ are taken must be used to yield **optimal** penalized estimation of the noisy data $\{\tau_i, f_i\}$ in nonparametric regression. However, since the points $\{\tau_i\}$ are chosen without any knowledge of the signal onset t_0 , they **cannot** be used as knots to estimate t_0 . Note that t_0 is a **variable** in our mathematical model. In addition, the number N of measurements is usually much larger than the realistic number n of knots. In order not to get too far away from nonparametric optimal penalized estimation, we therefore restrict our attention to the subspace $\mathbb{S}_{t_h,0}^k$ with basis $\{B_{k,t_h,-k+1}, \dots, B_{k,t_h,n-1}\}$ of $W_{2,0}^m$ where $k = 2m - 1$. That is, every spline subspace function in $\mathbb{S}_{t_h,0}^k$ is a spline series

$$S_k(t) = \sum_{j=-k+1}^{n-1} c_j B_{k,t_h,j}(t), \quad (60)$$

as discussed before. Here,

$$t_h: t_{-k+1} = \dots = t_{-1} = t_0 < t_1 < \dots < t_{n+k}, \quad (61)$$

with $0 \leq t_0 = c < d$ and $t_n = d$, is the knot sequence of the space $\mathbb{S}_{t_h,0}^k$, as discussed in the Introduction. Observe the effect that if n is chosen to be $N + 1$ and the τ_i 's are also equally spaced, then the best choice of t_0 will enable us to reproduce the nonparametric optimal penalized estimation. It must be emphasized that only odd degree splines should be used to retain the spirit of nonparametric optimization; hence, corresponding to $m = 1, 2$, degrees $k = 1$ and 3 (i.e. linear and cubic splines) will be most useful. For $g(t) = S_k(t)$ given by the spline series in Eq. (60), the functional $G_h(g, \lambda)$ in Eq. (50) that we are interested in minimizing becomes

$$\begin{aligned} G_h(S_k, \lambda) &= \|S_k - \mathbf{f}\|_2^2 + \lambda \int_c^d (S_k^{(m)}(t))^2 dt \\ &= \frac{1}{N} \sum_{i=1}^N \left[f_i - \sum_{j=-k+1}^{n-1} c_j B_{k,t_h,j}(\tau_i) \right]^2 w_i \\ &\quad + \lambda \sum_{\ell=-k+1}^{n-1} \sum_{j=-k+1}^{n-1} c_\ell c_j \int_c^d B_{k,t_h,j}^{(m)}(t) B_{k,t_h,\ell}^{(m)}(t) dt. \end{aligned} \quad (62)$$

The only difference of this from the minimization problem Eq. (52) is that we now minimize over all $S_k(t)$ in the subspace $\mathbb{S}_{t_h,0}^k$ instead of over all $g(t)$ the entire space $W_{2,0}^m$; in other words, the minimization is over all coefficients c_{-k+1}, \dots, c_{n-1} . Hence, as in Ref. 1 (pages 9-10), to find the spline series

$$S_k^*(t) = \sum_{j=-k+1}^{n-1} c_j^* B_{k,t_h,j}(t) \quad (63)$$

that satisfies

$$G_h(S_k^*, \lambda) = \hat{E}_k(h, \lambda), \quad (64)$$

where

$$\widehat{E}_k(h, \lambda) = \min_{S_k \in \mathbf{S}_{\mathbf{t}_h, 0}^k} G_h(S_k, \lambda), \quad (65)$$

we simply differentiate $G_h(S_k, \lambda)$ in Eq. (62) with respect to each of c_{-k+1}, \dots, c_{n-1} to yield the *normal equations*:

$$\sum_{j=-k+1}^{n-1} (a_{k,ij} + N\lambda b_{k,ij}^m) c_j^* = f_{k,i}, \quad i = -k+1, \dots, n-1, \quad (66)$$

where

$$a_{k,ij} = \sum_{\ell=1}^N B_{k,\mathbf{t}_h,i}(\tau_\ell) B_{k,\mathbf{t}_h,j}(\tau_\ell) w_\ell, \quad (67)$$

$$b_{k,ij} = \int_c^d B_{k,\mathbf{t}_h,i}^{(m)}(t) B_{k,\mathbf{t}_h,j}^{(m)}(t) dt, \quad (68)$$

and

$$\bar{f}_{k,i} = \sum_{\ell=1}^N f_\ell B_{k,\mathbf{t}_h,i}(\tau_\ell) w_\ell. \quad (69)$$

By setting

$$\tilde{A}_{k,h} = [a_{k,ij}]_{-k+1 \leq i, j \leq n-1} \quad (70)$$

$$B_{k,h}^0 = [b_{k,ij}]_{-k+1 \leq i, j \leq n-1} \quad (71)$$

and

$$\mathbf{c}^* = \begin{bmatrix} c_{-k+1}^* \\ \vdots \\ c_{n-1}^* \end{bmatrix}, \quad \bar{\mathbf{f}}_h = \begin{bmatrix} \bar{f}_{h,-k+1} \\ \vdots \\ \bar{f}_{h,n-1} \end{bmatrix}, \quad (72)$$

the matrix formulation of the normal equations in Eq. (66) becomes:

$$(\tilde{A}_{k,h} + N\lambda B_{k,h}^0) \mathbf{c}^* = \bar{\mathbf{f}}_h. \quad (73)$$

In computation, we first determine the $N \times (n+k-1)$ matrix

$$\tilde{B}_{k,h} = [B_{k,\mathbf{t}_h,j}(\tau_i)]_{\substack{-k+1 \leq i \leq N \\ -k+1 \leq j \leq n-1}}. \quad (74)$$

Then setting

$$\mathbf{W} = \begin{bmatrix} w_1 & & \mathbb{O} \\ & \ddots & \\ \mathbb{O} & & w_N \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}, \quad (75)$$

where $\{f_i\}$ is the noisy data taken at $\{\tau_i\}$, the normal equations in Eq. (66), or equivalently Eq. (73), become

$$(\tilde{B}_{k,h}^T W \tilde{B}_{k,h} + N\lambda B_{k,h}^0) \mathbf{c}^* = \tilde{B}_{k,h}^T W \mathbf{f}. \quad (76)$$

It should be remarked that the $(n+k-1) \times (n+k-1)$ matrix $B_{k,h}^0$ can be pre-computed as what we will do in the next sub-section. There, we will see that in general this matrix $B_{k,h}^0$ is fairly complicated. Since it has to be multiplied by the smoothing parameter λ which still has to be estimated, inverting the coefficient matrix of the unknown vector \mathbf{c}^* in Eq. (76) to find \mathbf{c}^* causes some complication. For this reason, we proposed [Eq. (27) on page 10 of Ref. 1] to replace the matrix $B_{k,h}^0$ by the identity matrix I ; that is, the modification

$$(\tilde{B}_{k,h}^T W \tilde{B}_{k,h} + N\lambda I) \mathbf{c}^* = \tilde{B}_{k,h}^T W \mathbf{f} \quad (77)$$

of Eq. (76) will also be considered.

Computation of the Matrix $B_{k,h}^0$

If we wish to use the original normal equations in Eq. (66), or equivalently (76), we must pre-compute the matrix $B_{k,h}^0$. For linear splines (i.e $m = k = 1$), this can be done easily. Indeed, by Eq. (2) for $k = 1$, we have $B'_{1,t_h,j}(t) = (1/h)N'_1[(1/h)(t - c) - j]$ where

$$N'(t) = \begin{cases} 1 & \text{for } 0 < t < 1 \\ -1 & \text{for } 1 < t < 2, \\ 0 & \text{otherwise} \end{cases}$$

so that

$$\begin{aligned} b_{1,ij} &= \int_c^d B'_{1,t_h,i}(t) B'_{1,t_h,j}(t) dt \\ &= \frac{1}{h} \int_0^n N'(t-i) N'(t-j) dt \\ &= \begin{cases} \frac{2}{h} & \text{for } i = j, \quad 0 \leq i \leq n-2, \\ \frac{1}{h} & \text{for } i = j, \quad i = n-1, \\ -\frac{1}{h} & \text{for } |i-j| = 1, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (78)$$

That is, the matrix $B_{1,h}^0$ in Eq. (76) for $k = 1$ is given by

$$B_{1,h}^0 = \frac{1}{h} \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & \ddots & & \bigcirc \\ & \ddots & \ddots & \ddots & \\ \bigcirc & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}. \quad (79)$$

For cubic splines (i.e. $k = 3, m = 2$), the computation is much more involved. We will again use Bernstein representations. Hence, from Figs. 2 and 3, and the corresponding expressions that follow the figures, the Bernstein representations of $\tilde{N}_3''(t)$, $\hat{N}_3''(t)$, and $N_3''(t)$ are piecewise linear polynomials given by the Bernstein representations in Fig. 4 below:

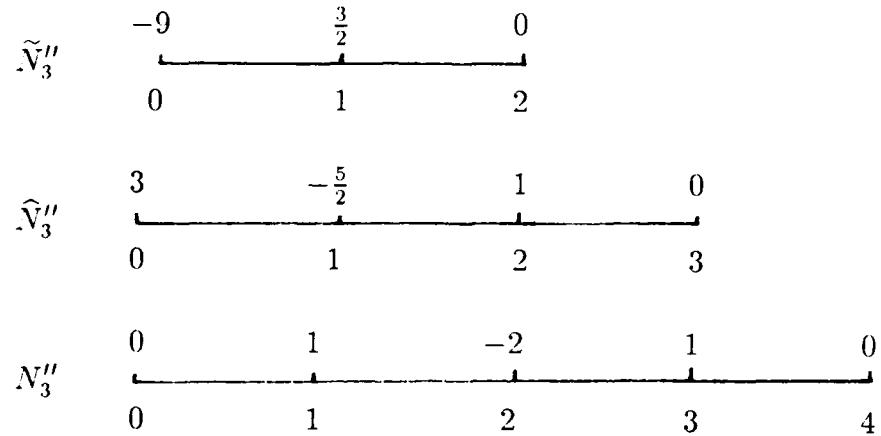


Fig. 4 - Bernstein representations of $\tilde{N}_3''(t)$, $\hat{N}_3''(t)$, and $N_3''(t)$

Hence, by applying the integration formula in Eq. (16), we have the following results:

(a) For $i = j = -2$, we have

$$\begin{aligned} b_{3;-2,-2} &= \frac{1}{h^3} \int_0^2 (\tilde{N}_3''(t))^2 dt \\ &= \frac{24}{h^3} = \frac{288}{2 \times 3!h^3}. \end{aligned}$$

(b) For $i = -2, j = -1$ or $i = -1, j = -2$,

$$\begin{aligned} b_{3;-2,-1} = b_{3;-1,-2} &= \frac{1}{h^3} \int_0^2 \tilde{N}_3''(t) \hat{N}_3''(t) dt \\ &= -\frac{27}{4h^3} = -\frac{81}{2 \times 3!h^3}. \end{aligned}$$

(c) For $i = -2, j = 0$ or $i = 0, j = -2$, we have

$$\begin{aligned} b_{3;-2,0} = b_{3,0,-2} &= \frac{1}{h^3} \int_0^2 \tilde{N}_3''(t) N_3''(t) dt \\ &= -\frac{1}{h^3} = -\frac{12}{2 \times 3!h^3}. \end{aligned}$$

(d) For $i = -2, j = 1$ or $i = 1, j = -2$, we have

$$\begin{aligned} b_{3;-2,1} &= b_{3,1,-2} = \frac{1}{h^3} \int_1^2 \tilde{N}_3''(t) N_3(t-1) dt \\ &= \frac{1}{4h^3} = \frac{3}{2 \times 3!h^3}. \end{aligned}$$

(e) For $i = -2, j \geq 2$, or $i \geq 2, j = -2$, we have

$$b_{3;-2,j} = b_{3;i,-2} = 0.$$

(f) For $i = -1, j = -1$, we have

$$\begin{aligned} b_{3;-1,-1} &= \frac{1}{h^3} \int_0^3 (\tilde{N}_3''(t))^2 dt \\ &= \frac{9}{2h^3} = \frac{54}{2 \times 3!h^3}. \end{aligned}$$

(g) For $i = -1, j = 0$ or $i = 0, j = -1$, we have

$$\begin{aligned} b_{3;-1,0} &= b_{3,0,-1} = \frac{1}{h^3} \int_0^3 \tilde{N}_3''(t) N_3''(t) dt \\ &= -\frac{4}{3h^3} = -\frac{16}{2 \times 3!h^3}. \end{aligned}$$

(h) For $i = -1, j = 1$ or $i = 1, j = -1$, we have

$$\begin{aligned} b_{3;-1,1} &= b_{3,1,-1} = \frac{1}{h^3} \int_1^3 \tilde{N}_3''(t) N_3''(t-1) dt \\ &= -\frac{1}{12h^3} = -\frac{1}{2 \times 3!h^3}. \end{aligned}$$

(i) For $i = -1, j = 2$ or $i = 2, j = -1$, we have

$$\begin{aligned} b_{3;-1,2} &= b_{3,2,-1} = \frac{1}{h^3} \int_2^3 \tilde{N}_3''(t) N_3''(t-2) dt \\ &= \frac{1}{6h^3} = \frac{2}{2 \times 3!h^3}. \end{aligned}$$

(j) For $i = -1, j \geq 3$ or $i \geq 3, j = -1$, we have

$$b_{3;-1,j} = b_{3;i,-1} = 0.$$

(k) For $i = j$ and $i = 0, \dots, n-4$, we have

$$b_{3,ii} = \frac{1}{h^3} \int_0^4 (N_3''(t))^2 dt = \frac{8}{3h^3} = \frac{32}{2 \times 3!h^3}.$$

(l) For $i = j = n-3$, we have

$$b_{3;n-3,n-3} = \frac{1}{h^3} \int_0^3 (N_3''(t))^2 dt = \frac{7}{3h^3} = \frac{28}{2 \times 3!h^3}.$$

(m) For $i = j = n - 2$, we have

$$b_{3;n-2,n-2} = \frac{1}{h^3} \int_0^2 (N_3''(t))^2 dt = \frac{4}{3h^3} = \frac{16}{2 \times 3!h^3}.$$

(n) For $i = j = n - 1$, we have

$$b_{3;n-1,n-1} = \frac{1}{h^3} \int_0^1 (N_3''(t))^2 dt = \frac{1}{3h^3} = \frac{4}{2 \times 3!h^3}.$$

(o) For $i = 0, \dots, n-4, j = i+1$, or $i = j+1, j = 0, \dots, n-4$, we have

$$\begin{aligned} b_{3;i,i+1} = b_{3;j+1,j} &= \frac{1}{h^3} \int_1^4 N_3''(t)N_3''(t-1)dt \\ &= -\frac{3}{2h^3} = -\frac{18}{2 \times 3!h^3}. \end{aligned}$$

(p) For $i = n-3, j = i+1 = n-2$, or $i = n-2, j = n-1$, we have

$$\begin{aligned} b_{3;n-3,n-2} = b_{3;n-2,n-3} &= \frac{1}{h^3} \int_1^3 N_3''(t)N_3''(t-1)dt \\ &= -\frac{1}{h^3} = -\frac{12}{2 \times 3!h^3}. \end{aligned}$$

(q) For $i = n-2, j = i+1 = n-1$, or $i = n-1, j = n-2$, we have

$$\begin{aligned} b_{3;n-2,n-1} = b_{3;n-1,n-2} &= \frac{1}{h^3} \int_1^2 N_3''(t)N_3''(t-1)dt \\ &= -\frac{1}{2h^3} = -\frac{1}{2 \times 3!h^3}. \end{aligned}$$

(r) For $i = 0, \dots, n-4$ and $j = i+2$; or $i = j+2$ and $j = 0, \dots, n-4$, we have

$$\begin{aligned} b_{3;i,j} &= \frac{1}{h^3} \int_2^4 N_3''(t)N_3''(t-2)dt \\ &= \frac{1}{6h^3}(2-2+2-2) = 0. \end{aligned}$$

(s) For $i = n-3$ and $j = n-1$, or $i = n-1$ and $j = n-3$, we have

$$b_{3;i,j} = \frac{1}{h^3} \int_2^3 N_3''(t)N_3''(t-2)dt = \frac{1}{6h^3}(2-2) = 0.$$

(t) For $i = 0, \dots, n-4$ and $j = i+3$, or $i = j+3$ and $j = 0, \dots, n-4$, we have

$$b_{3;i,j} = \frac{1}{h^3} \int_3^4 N_3''(t)N_3''(t-3)dt = \frac{1}{6h^3} = \frac{2}{2 \times 3!h^3}.$$

(u) For $|i-j| \geq 4$, we have $b_{3;i,j} = 0$.

Hence, summarizing the results in (a)-(u), we have obtained the $(n+2)$ by $(n+2)$ matrix

$$B_{3,h}^0 = \frac{1}{2 \times 3!h^3} \begin{bmatrix} b_{11}^0 & b_{12}^0 & 0 \\ b_{21}^0 & b_{22}^0 & b_{23}^0 \\ 0 & b_{32}^0 & b_{33}^0 \end{bmatrix} \quad (80)$$

where

$$b_{11}^0 = \begin{bmatrix} 288 & -81 \\ -81 & 54 \end{bmatrix}, \quad b_{12}^0 = \begin{bmatrix} -12 & 3 & 0 & 0 & \dots & 0 \\ -16 & -1 & 2 & 0 & \dots & 0 \end{bmatrix},$$

$$b_{22}^0 = \begin{bmatrix} 32 & -18 & 0 & 2 & \dots & \textcircled{O} \\ -18 & \ddots \\ 0 & \ddots & 2 \\ 2 & \ddots & 0 \\ \textcircled{O} & \ddots & -18 \\ & & 2 & 0 & -18 & 32 & & & & & & \end{bmatrix},$$

$$b_{23}^0 = \begin{bmatrix} \textcircled{O} & & \\ 2 & 0 & 0 \\ 0 & 2 & 0 \\ -18 & 0 & 2 \end{bmatrix}, \quad b_{33}^0 = \begin{bmatrix} 28 & -12 & 0 \\ -12 & 16 & -6 \\ 0 & -6 & 4 \end{bmatrix}$$

are 2×2 , $2 \times (n-3)$, $(n-3) \times (n-3)$, $(n-3) \times 3$, and 3×3 blocks and $b_{21}^0 = b_{12}^{0T}$, $b_{32}^0 = b_{23}^{0T}$.

In general, to compute $B_{k,h}^0$ for $k = 5, 7, \dots$, one has first to determine the corresponding B-splines $N_{k,\ell}(t)$ with ℓ stacked knots at the origin, $\ell = 2, \dots, k-1$ [where $N_{k,1}(t) \equiv N_k(t)$], then compute their m th derivatives where $m = (k+1)/2$, and finally evaluate the integrals:

$$b_{k,ij} = \frac{1}{h^k} \int_0^{k+1} N_{k,-i+1}^{(m)}(t) N_{k,-j+1}^{(m)}(t) dt \quad (81)$$

for $-k+1 \leq i, j \leq -1$;

$$b_{k,ij} = \frac{1}{h^k} \int_0^n N_{k,-i+1}^{(m)}(t) N_k^{(m)}(t-j) dt \quad (82)$$

for $-k+1 \leq i \leq -1$ and $0 \leq j \leq n-1$, or $0 \leq i \leq n-1$ and $-k+1 \leq j \leq -1$; and

$$b_{k,ij} = \frac{1}{h^k} \int_0^n N_k^{(m)}(t-i) N_k^{(m)}(t-j) dt \quad (83)$$

for $0 \leq i, j \leq n-1$. Of course, the formula in Eq. (16) should be used to facilitate the integration procedures.

The Generalized Cross-Validation Function

Let us now return to the normal equations in Eq. (76) and the modification in Eq. (77) with $B_{k,h}^0$ replaced by the identity matrix I. Note that the matrix $\hat{B}_{k,h}$

$[B_{k,t_h,j}(\tau_i)]$ is simple to evaluate and the weight matrix W and data vector \mathbf{f} are already predetermined and given. Hence, it is simple to determine $\mathbf{c}^* = \mathbf{c}^*(h, \lambda)$ for any fixed values of h and λ by using the Moore-Penrose pseudoinverse in Ref. 1 (pages 16-20). We will always fix $h > 0$ in the following discussion. The objective is to determine an optimal value of the smoothing parameter λ in terms of the noise variances of the given data $\mathbf{f} = \{f_i\}$. Let $\mathbf{c}^* = \mathbf{c}^*(\lambda) = [c_{-k+1}^* \dots c_{n-1}^*]^T$ be the solution of Eq. (76) or Eq. (77) using the Moore-Penrose pseudoinverse

$$M(\lambda) = \begin{cases} (\tilde{B}_{k,h}^T W \tilde{B}_{k,h} + N\lambda B_{k,h}^0)^+ & \text{for Eq. (76)} \\ (\tilde{B}_{k,h}^T W \tilde{B}_{k,h} + N\lambda I)^+ & \text{for Eq. (77)} \end{cases} \quad (84)$$

in the sense that $\mathbf{c}^*(\lambda) = M(\lambda) \tilde{B}_{k,h}^T W \mathbf{f}$ is the unique solution of Eq. (76) or (77) with minimum ℓ^2 -norm. (See page 16 in Ref. 1). In other words,

$$S^*(t; \lambda) = \sum_{j=-k+1}^{n-1} c_j^*(\lambda) B_{k,t_h,j}(t) \quad (85)$$

is the (optimal) *penalized least-squares estimator* of the noisy data $\mathbf{f} = \{f_i\}$ taken at the sample nodes $\{\tau_i\}$. To estimate the optimal smoothing parameter λ , we investigate how well $S^*(t; \lambda)$ approximates the *actual* (unknown) signal $\{f^*(\tau_i)\}$ [when the noise $\{\varepsilon_i\}$ is removed from the measured data $\{f_i\}$; see Eq. (46) for the definition of $f^*(\tau_i)$]; that is, we are interested in the size of the quantity

$$T_{\mathbf{f}}(\lambda) = \frac{1}{N} \sum_{i=1}^N w_i (S^*(\tau_i; \lambda) - f^*(\tau_i))^2. \quad (86)$$

Since $T_{\mathbf{f}}(\lambda)$ depends on the noisy data \mathbf{f} , we must, in fact, investigate its mathematical expectation $\mathcal{E}T_{\mathbf{f}}(\lambda)$. It seems then very reasonable to characterize the *optimal smoothing parameter* $\lambda^* = \lambda_N^*$ as one that minimizes $\mathcal{E}T_{\mathbf{f}}(\lambda)$, namely:

$$\mathcal{E}T_{\mathbf{f}}(\lambda_N^*) = \min_{\lambda \geq 0} \mathcal{E}T_{\mathbf{f}}(\lambda). \quad (87)$$

However, this is an impossible task since the actual signal $\{f^*(\tau_i)\}$ in the definition of $T_{\mathbf{f}}(\lambda)$ is unknown. For this reason, the alternative quantity

$$\tilde{T}_{\mathbf{f}}(\lambda) = \frac{1}{N} \sum_{i=1}^N w_i (S^*(\tau_i; \lambda) - f_i)^2 \quad (88)$$

which measure the error in approximating the data $\mathbf{f} = \{f_i\}$ by the optimal penalized least-squares estimator $S^*(t; \lambda)$ must be taken into consideration. In fact, it will be seen that a weighted quantity of $\tilde{T}_{\mathbf{f}}(\lambda)$ given by

$$V_N(\lambda) = \frac{\tilde{T}_{\mathbf{f}}(\lambda)}{(1 - L_N(\lambda))^2}, \quad (89)$$

where

$$L_N(\lambda) = \frac{1}{N} \text{Trace}[\tilde{B}_{k,h} M(\lambda) \tilde{B}_{k,h}^T W] \quad (90)$$

with $M(\lambda)$ defined in Eq. (84), shall give some information on the optimal λ_N^* . The utility of the weight $(1 - L_N(\lambda))^{-2}$ will be clear later. The function $V_N(\lambda)$ is called the *generalized cross-validation function* for the data \mathbf{f} , and the matrix

$$\mathcal{J}(\lambda) = \tilde{\mathbf{B}}_{k,h} M(\lambda) \tilde{\mathbf{B}}_{k,h}^T W \quad (91)$$

in Eq. (90) is called the *influence matrix*. (See [3-5,8]).

Let $\tilde{\lambda} = \tilde{\lambda}_N$ be the λ that minimizes the generalized cross-validation function in the sense that

$$V_N(\tilde{\lambda}_N) = \min_{\lambda \geq 0} V_N(\lambda). \quad (92)$$

In the following we will show that for large data samples (i.e. when N is a very large value), $\tilde{\lambda}_N$ can be used as a good estimate of the optimal smoothing parameter λ_N^* in the sense that *the error in approximating the actual signal $\{f^*(\tau_i)\}$ by using the optimal least-squares estimator $S^*(t; \tilde{\lambda}_N)$ is asymptotically the same as that by using $S^*(t; \lambda_N^*)$ with the optimal λ_N^** , namely:

$$\lim_{N \rightarrow \infty} \frac{\mathcal{E}T_{\mathbf{f}}(\tilde{\lambda}_N)}{\mathcal{E}T_{\mathbf{f}}(\lambda_N^*)} = 1. \quad (93)$$

(Note that the limit actually exists.)

Before we attempt to prove Eq. (93), we need a better understanding of $\mathcal{E}T_{\mathbf{f}}(\lambda)$. First, by using the notations

$$\mathbf{s}^0(\lambda) = \begin{bmatrix} S^*(\tau_1; \lambda) \\ \vdots \\ S^*(\tau_N; \lambda) \end{bmatrix} \quad (94)$$

and

$$\mathbf{f}^* = \begin{bmatrix} f^*(\tau_1) \\ \vdots \\ f^*(\tau_N) \end{bmatrix}, \quad (95)$$

we have, from Eq. (86),

$$\begin{aligned} NT_{\mathbf{f}}(\lambda) &= \sum_{i=1}^N w_i (S^*(\tau_i, \lambda) - f^*(\tau_i))^2 \\ &= (\mathbf{s}^0(\lambda) - \mathbf{f}^*)^T W (\mathbf{s}^0(\lambda) - \mathbf{f}^*). \end{aligned} \quad (96)$$

But since $\mathbf{s}^0(\lambda) = \tilde{\mathbf{B}}_{k,h} \mathbf{c}^*(\lambda) = \tilde{\mathbf{B}}_{k,h} M(\lambda) \tilde{\mathbf{B}}_{k,h}^T W \mathbf{f} = \mathcal{J}(\lambda) \mathbf{f}$ [see Eqs. (84) and (91)], we have

$$\begin{aligned}
NT_{\mathbf{f}}(\lambda) &= (\mathcal{J}(\lambda)\mathbf{f} - \mathbf{f}^*)^T W (\mathcal{J}(\lambda)\mathbf{f} - \mathbf{f}^*) \\
&= [\mathcal{J}(\lambda)(\mathbf{f}^* + \varepsilon) - \mathbf{f}^*]^T W [\mathcal{J}(\lambda)(\mathbf{f}^* + \varepsilon) - \mathbf{f}^*] \\
&= [\mathcal{J}(\lambda)\mathbf{f}^* - \mathbf{f}^*]^T W [\mathcal{J}(\lambda)\mathbf{f}^* - \mathbf{f}^*] \\
&\quad + \varepsilon^T \mathcal{J}^T(\lambda) W [\mathcal{J}(\lambda)\mathbf{f}^* - \mathbf{f}^*] + [\mathcal{J}(\lambda)\mathbf{f}^* - \mathbf{f}^*]^T W \mathcal{J}(\lambda) \varepsilon \\
&\quad + \varepsilon^T \mathcal{J}^T(\lambda) W \mathcal{J}(\lambda) \varepsilon
\end{aligned} \tag{97}$$

where $\varepsilon = [\varepsilon_1 \dots \varepsilon_N]^T$. Hence, it follows from the assumptions $\mathcal{E}(\varepsilon_i) = 0$ and $\mathcal{E}(\varepsilon_i \varepsilon_j) = \sigma_i^2 \delta_{ij}$ on the noise that

$$\begin{aligned}
\mathcal{E}T_{\mathbf{f}}(\lambda) &= \frac{1}{N} [\mathcal{J}(\lambda)\mathbf{f}^* - \mathbf{f}^*]^T W [\mathcal{J}(\lambda)\mathbf{f}^* - \mathbf{f}^*] \\
&\quad + \frac{1}{N} \text{Trace}[\mathcal{J}^T(\lambda) W \Sigma^2 \mathcal{J}(\lambda)] \\
&= T_{\mathbf{f}^*}(\lambda) + \left(\sum_{i=1}^N \sigma_i^{-2} \right)^{-1} \text{Trace}[\mathcal{J}^T(\lambda) \mathcal{J}(\lambda)]
\end{aligned} \tag{98}$$

where $\Sigma^2 = \text{diag}(\sigma_1^2, \dots, \sigma_n^2)$ has been combined with the weight matrix W by using the choice of W in Eq. (49) to yield the constant factor $(\sum_{i=1}^N \sigma_i^{-2})^{-1}$ of the trace, and $T_{\mathbf{f}^*}$ differs from $T_{\mathbf{f}}$ in Eq. (86) in that the optimal estimator $S^*(\tau_i, \lambda)$ in Eq. (86) must now be replaced by the corresponding one that estimates the noiseless signal \mathbf{f}^* . On the other hand, by the same derivation as above, we also have

$$\begin{aligned}
\mathcal{E}\tilde{T}_{\mathbf{f}}(\lambda) &= \mathcal{E}T_{\mathbf{f}}(\lambda) + \frac{1}{N} \left(\text{Trace } W \Sigma^2 - 2 \text{Trace } W \Sigma^2 \mathcal{J}(\lambda) \right) \\
&= \mathcal{E}T_{\mathbf{f}}(\lambda) + \left(\sum_{i=1}^N \sigma_i^{-2} \right)^{-1} [N - 2 \text{Trace } \mathcal{J}(\lambda)];
\end{aligned} \tag{99}$$

so that

$$\begin{aligned}
\mathcal{E}T_{\mathbf{f}}(\lambda) &= \left(1 - \frac{1}{N} \text{Trace } \mathcal{J}(\lambda) \right)^2 \mathcal{E}V_N(\lambda) \\
&\quad + \left(\sum_{i=1}^N \sigma_i^{-2} \right)^{-1} [2 \text{Trace } \mathcal{J}(\lambda) - N].
\end{aligned} \tag{100}$$

In the following estimates, in order to clarify the presentation, we will use the abbreviations:

$$a = \frac{1}{N} \text{Trace } \mathcal{J}(\lambda), \quad b = \text{Trace}[\mathcal{J}^T(\lambda) \mathcal{J}(\lambda)], \quad c = \left(\sum_{i=1}^N \sigma_i^{-2} \right)^{-1}. \tag{101}$$

Hence, by using Eqs. (98) and (100), we have

$$\begin{aligned}
 & \left| \frac{\mathcal{E}T_{\mathbf{f}}(\lambda) - \mathcal{E}V_N(\lambda) + cN}{\mathcal{E}T_{\mathbf{f}}} \right| \\
 &= \left| \frac{T_{\mathbf{f}^*}(\lambda) + bc - (1-a)^{-2}[T_{\mathbf{f}^*}(\lambda) + bc - 2acN + cN] + cN}{T_{\mathbf{f}^*}(\lambda) + bc} \right| \\
 &= \left| \frac{-2a + a^2}{(1-a)^2} + \frac{(2acN - cN) + (cN - 2acN + ca^2N)}{(T_{\mathbf{f}^*}(\lambda) + bc)(1-a)^2} \right| \\
 &= \left| a - 2 + \frac{cN a}{T_{\mathbf{f}^*} + bc} \right| \cdot \frac{|a|}{(1-a)^2}.
 \end{aligned} \tag{102}$$

By setting

$$\delta_N(\lambda) = \frac{|a|}{(1-a)^2} \left[|2-a| + \frac{1}{b} N |a| \right] \tag{103}$$

and observing that $T_{\mathbf{f}^*}(\lambda) \geq 0$, we have

$$\left| \frac{\mathcal{E}T_{\mathbf{f}}(\lambda) - \mathcal{E}V_N(\lambda) + cN}{\mathcal{E}T_{\mathbf{f}}(\lambda)} \right| \leq \delta_N(\lambda). \tag{104}$$

Now, by the definition of $\tilde{\lambda} = \tilde{\lambda}_N$, we have $V_N(\tilde{\lambda}) \leq V_N(\lambda^*)$, so that $\mathcal{E}V_N(\tilde{\lambda}) \leq \mathcal{E}V_N(\lambda^*)$, and hence Eq. (104) implies:

$$\begin{aligned}
 (\mathcal{E}T_{\mathbf{f}}(\tilde{\lambda}))(1 - \delta_N(\tilde{\lambda})) &\leq \mathcal{E}V_N(\tilde{\lambda}) - cN \\
 &\leq \mathcal{E}V_N(\lambda^*) - cN \\
 &\leq (\mathcal{E}T_{\mathbf{f}}(\lambda^*))(1 + \delta_N(\tilde{\lambda})).
 \end{aligned} \tag{105}$$

In addition, by the definition of $\lambda^* = \lambda_N^*$, we have $\mathcal{E}T_{\mathbf{f}}(\lambda^*) \leq \mathcal{E}T_{\mathbf{f}}(\tilde{\lambda})$. Therefore, it follows that

$$1 \leq \frac{\mathcal{E}T_{\mathbf{f}}(\tilde{\lambda}_N)}{\mathcal{E}T_{\mathbf{f}}(\lambda_N^*)} \leq \frac{1 + \delta_N(\tilde{\lambda}_N)}{1 - \delta_N(\tilde{\lambda}_N)}. \tag{106}$$

These inequalities will yield the limit result in Eq. (93) once we establish $\delta_N(\tilde{\lambda}_N) \rightarrow 0$; but in view of its definition in Eq. (103) and the definition of a in Eq. (101), we indeed have $\delta_N(\lambda) \rightarrow 0$ uniformly in $\lambda > 0$ if we can show that $\text{Trace } \mathcal{J}(\lambda)$ is uniformly bounded. This fact will be verified in the next section. We emphasize once again that Eq. (93) assures us that $\tilde{\lambda} = \tilde{\lambda}_N$ provides a good estimate for the optimal smoothing parameter $\lambda^* = \lambda_N^*$.

ALGORITHMS FOR NEAR-OPTIMAL ESTIMATION

In estimation of the signal onset of an underwater acoustic signal $\mathbf{f}^* = \{f^*(\tau_i)\}$ from the noisy data $\mathbf{f} = \{f_i\}$ where $f_i = f^*(\tau_i) + \varepsilon_i$, $i = 1, \dots, N$, by applying the method of penalized least-squares regression as discussed in the section entitled Penalized Least-Squares Estimation, one of the main difficulties is the determination of the optimal smoothing parameter λ . For a large quantity N of data samples, a good estimate of the optimal $\lambda = \lambda^*$ is given by $\tilde{\lambda} = \tilde{\lambda}_N$ which minimizes the generalized cross-validation function $V_N(\lambda)$ defined in Eq. (89). Since the error functional $\tilde{T}_{\mathbf{f}}(\lambda)$ can be

easily computed for each fixed value of λ (and of h) by first computing the pseudoinverse $M(\lambda)$ in Eq. (84), the only factor of $V_N(\lambda)$ that remains to be computed is the trace of the influence matrix $\mathcal{J}(\lambda)$ defined in Eq. (91). In the following, we will simplify $V_N(\lambda)$ so that $\tilde{T}_f(\lambda)$ and Trace $\mathcal{J}(\lambda)$ can be computed simultaneously, so that $\tilde{\lambda} = \tilde{\lambda}_N$ can be determined more efficiently.

Two Formulations of the Generalized Cross-Validation Function

We will first write the generalized cross-validation function $V_N(\lambda)$ in a more manageable form. To do so, we need some notations. First, we can assume, without loss of generality, that each weight w_i is non-zero, since the data information f_i for the zero weight w_i can be dropped. Hence, we may use the notations:

$$W^{\frac{1}{2}} = \begin{bmatrix} w_1^{\frac{1}{2}} & & & \\ & \ddots & & \\ & & \ddots & \\ \textcircled{O} & & & w_N^{\frac{1}{2}} \end{bmatrix}, \quad W^{-\frac{1}{2}} = \begin{bmatrix} w_1^{-\frac{1}{2}} & & & \\ & \ddots & & \\ & & \ddots & \\ \textcircled{O} & & & w_N^{-\frac{1}{2}} \end{bmatrix}. \quad (107)$$

The positive definite banded symmetric matrix $B_{k,h}^0$ defined in Eqs. (71) and (68) can also be treated in the same fashion by first diagonalizing the matrix and then taking the positive square roots of its eigenvalues. Hence, $(B_{k,h}^0)^{1/2}$ is also positive definite and symmetric and $(B_{k,h}^0)^{1/2}(B_{k,h}^0)^{1/2} = B_{k,h}^0$. Recall that by using Eq. (84), the influence matrix $\mathcal{J}(\lambda)$ can be transformed to a nonnegative definite symmetric matrix

$$\begin{aligned} \tilde{\mathcal{J}}(\lambda) &= W^{\frac{1}{2}} \mathcal{J}(\lambda) W^{-\frac{1}{2}} \\ &= W^{\frac{1}{2}} \tilde{B}_{k,h} [\tilde{B}_{k,h}^T W \tilde{B}_{k,h} + N\lambda B_{k,h}^0]^{-1} \tilde{B}_{k,h}^T W^{\frac{1}{2}} \end{aligned} \quad (108)$$

where the matrix $B_{k,h}^0$ should be replaced by the identity matrix if the modification Eq. (77) of the normal equation in Eq. (76) is to be studied (to simplify the computational procedure). In any case, by setting

$$X = \begin{cases} W^{\frac{1}{2}} \tilde{B}_{k,h} (B_{k,h}^0)^{-\frac{1}{2}} & \text{for Eq. (76)} \\ W^{\frac{1}{2}} \tilde{B}_{k,h} & \text{for Eq. (77)} \end{cases} \quad (109)$$

we may write

$$\tilde{\mathcal{J}}(\lambda) = X (X^T X + N\lambda I)^{-1} X^T. \quad (110)$$

[We remark that the matrix $B_{k,h}^0$ may be singular. In this case, we recommend the model in Eq. (77) over the model in Eq. (76).] On the other hand, by setting

$$\tilde{\mathbf{f}} = W^{\frac{1}{2}} \mathbf{f} \quad (111)$$

and noting

$$\text{Trace } \tilde{\mathcal{J}}(\lambda) = \text{Trace } \mathcal{J}(\lambda) \quad (112)$$

we may write

$$\begin{aligned}
 V_N(\lambda) &= \frac{\frac{1}{N}(\mathcal{J}(\lambda)\mathbf{f} - \mathbf{f})^T W(\mathcal{J}(\lambda)\mathbf{f} - \mathbf{f})}{(1 - \frac{1}{N} \text{Trace } \mathcal{J}(\lambda))^2} \\
 &= \frac{\frac{1}{N} \|W^{\frac{1}{2}}(\mathcal{J}(\lambda) - I)\mathbf{f}\|_{\ell^2}^2}{\frac{1}{N^2}(\text{Trace}[I - \mathcal{J}(\lambda)])^2},
 \end{aligned} \tag{113}$$

or equivalently,

$$V_N(\lambda) = \frac{\frac{1}{N} \|(I - \tilde{\mathcal{J}}(\lambda))\tilde{\mathbf{f}}\|_{\ell^2}^2}{\frac{1}{N^2}(\text{Trace}[I - \tilde{\mathcal{J}}(\lambda)])^2}, \tag{114}$$

where the non-weighted ℓ^2 norm [i.e. $\|(x_1, \dots, x_N)\|_{\ell^2}^2 = x_1^2 + \dots + x_N^2$] is used. (Note the similarity with the GCV function studied in Golub, Heath, and Wahba [8]). Let $s_1 \geq \dots \geq s_\ell > s_{\ell+1} = \dots = s_N = 0$ be the singular values of X ; that is,

$$s_1^2 \geq \dots \geq s_\ell^2 > s_{\ell+1}^2 = \dots = s_N^2 = 0 \tag{115}$$

are the eigenvalues of $X^T X$. Then by using the normalized eigenvectors of $X^T X$ corresponding to these eigenvalues, we can form two unitary matrices U and V (i.e. $UU^T = U^T U = I_N$ and $VV^T = V^T V = I_{n+k-1}$ where I_N and I_{n+k-1} are identity matrices) such that

$$X = U\Gamma V^T \tag{116}$$

where

$$\Gamma = \begin{bmatrix} s_1 & & \circ & & \vdots & & \\ & \ddots & & & \vdots & & \\ \circ & & s_\ell & & \vdots & & \circ \\ \dots & \dots & \dots & \dots & & & \\ & \circ & & & \vdots & & \circ \end{bmatrix}. \tag{117}$$

Note that $s_{\ell+1} = \dots = s_N = 0$ and

$$\Gamma^+ = \begin{bmatrix} s_1^{-1} & & \circ & & \vdots & & \\ & \ddots & & & \vdots & & \\ \circ & & s_\ell^{-1} & & \vdots & & \circ \\ \dots & \dots & \dots & \dots & & & \\ & \circ & & & \vdots & & \circ \end{bmatrix}. \tag{118}$$

(See [1], pages 18-22). Hence,

$$(X^T X + N\lambda I)^+ = V \begin{bmatrix} \frac{1}{s_1^2 + \lambda N} & & \circ & & \vdots & & \\ & \ddots & & \vdots & & & \circ \\ \circ & & \frac{1}{s_t^2 + \lambda N} & & \vdots & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ & & & \frac{1}{\lambda N} & & \circ & \\ \circ & & & \vdots & & \ddots & \\ & & & \circ & & & \frac{1}{\lambda N} \end{bmatrix} V^T \quad (119)$$

In addition, we have, from Eqs. (110), (116), and (119),

$$\tilde{\mathcal{J}}(\lambda) = U \begin{bmatrix} \frac{s_1^2}{s_1^2 + \lambda N} & & \circ & & \vdots & & \\ & \ddots & & \vdots & & & \circ \\ \circ & & \frac{s_t^2}{s_t^2 + \lambda N} & & \vdots & & \circ \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \circ & & & \vdots & & \circ & \end{bmatrix} U^T \quad (120)$$

and

$$I - \tilde{\mathcal{J}}(\lambda) = U \begin{bmatrix} \frac{\lambda N}{s_1^2 + \lambda N} & & \circ & & \vdots & & \\ & \ddots & & \vdots & & & \circ \\ \circ & & \frac{\lambda N}{s_t^2 + \lambda N} & & \vdots & & \circ \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \circ & & & \vdots & & \circ & \\ & & & \vdots & & \ddots & \\ & & & \circ & & & 1 \end{bmatrix} U^T. \quad (121)$$

Therefore, by using the transformation:

$$\hat{\mathbf{f}} = \begin{bmatrix} \hat{f}_1 \\ \vdots \\ \hat{f}_N \end{bmatrix} = U^T \tilde{\mathbf{f}} = U^T W^{\frac{1}{2}} \mathbf{f} \quad (122)$$

where \mathbf{f} is the original noisy data, we have

$$\|(I - \tilde{\mathcal{J}}(\lambda))\tilde{\mathbf{f}}\|_{l^2}^2 = \sum_{i=1}^N \left(\frac{N\lambda}{s_i^2 + N\lambda} \right)^2 \hat{f}_i^2. \quad (123)$$

In addition, it is clear that if D is diagonal and U is unitary, then the eigenvalues of UDU^T are the same as those of D , namely: the diagonal elements of D . Since the trace of a square matrix is the sum of all its eigenvalues, we have, from Eq. (121), that

$$\text{Trace}(I - \tilde{\mathcal{J}}(\lambda)) = \sum_{i=1}^{\ell} \frac{\lambda N}{s_i^2 + \lambda N} + N - \ell. \quad (124)$$

Putting Eqs. (123) and (124) into Eq. (114), we have

$$V_N(\lambda) = \sum_{i=1}^{\ell} \left(\frac{\hat{f}_i}{s_i^2 + N\lambda} \right)^2 \Big/ \left(\frac{1}{N} \left[\sum_{i=1}^{\ell} \frac{1}{s_i^2 + N\lambda} + \frac{N - \ell}{N\lambda} \right]^2 \right). \quad (125)$$

This formula is also similar to Eq. (2.3) given in Ref. 8.

Both formulations of Eqs. (114) and (125) of $V_N(\lambda)$ using the two different transformations of the noisy data $\tilde{\mathbf{f}} = W^{\frac{1}{2}}\mathbf{f}$ and $\hat{\mathbf{f}} = U^T W^{\frac{1}{2}}\mathbf{f}$, respectively, will be useful for computing the near optimal smoothing parameter $\tilde{\lambda} = \tilde{\lambda}_N$.

For the moment, let us apply Eq. (120) to study the truth of Eq. (93). From Eqs. (106) and (108), we have already noted that Eq. (93) is a consequence of the uniform boundedness of $\text{Trace } \tilde{\mathcal{J}}(\lambda)$ as a function of N for $\lambda \geq \lambda_0 > 0$ where λ_0 is arbitrary. By Eq. (120), we have

$$\text{Trace } \tilde{\mathcal{J}}(\lambda) = \sum_{i=1}^{\ell} \frac{s_i^2}{s_i^2 + \lambda N}. \quad (126)$$

Hence, for any fixed $\lambda_0 > 0$, if we can show that $s_i^2 \leq M$ for all $i = 1, \dots, \ell$ and all N , then

$$0 < \text{Trace } \tilde{\mathcal{J}}(\lambda) \leq \frac{NM}{\lambda_0 N - M} < \frac{2M}{\lambda_0} < \infty \quad (127)$$

for all large N . However, since s_i^2 are the eigenvalues of $X^T X$, it follows from Gershgorin's theorem, that

$$\max_i s_i^2 \leq \max_i \sum_{j=1}^N |a_{ij}| \quad (128)$$

where $X^T X = [a_{ij}]$. The boundedness of the sum of the absolute values along each row can be verified by using the properties of B -splines and Eq. (109); this is easily seen if the modified normal equation in Eq. (77) is used since $X^T X = \tilde{B}_{kh}^T W \tilde{B}_{kh}$ is banded and all nonzero entries are positive. A different but much more complicated proof can be found in Refs. 5 and 9.

Formulation of the Error Functional

In determining the time-of-arrival $t_0 = d - nh^*$, we must determine h^* which is the minimum number in the set $H = \{\tilde{h}\}$, where each \tilde{h} minimizes the error functional $\hat{E}_k(h, \lambda)$ for each fixed value of $\lambda > 0$. Recall that $\hat{E}_k(h, \lambda)$ is the error of optimal penalized estimation of any given noisy data \mathbf{f} from the spline space $\mathbb{S}_{t_h, 0}^k$ as defined in Eq.

(65). We must now give a computationally feasible formulation of $\widehat{E}_h(h, \lambda)$. Our aim is to arrive at a formula which allows us to determine \tilde{h} without getting into the trouble of computing \mathbf{c}^* that uniquely determine the last estimator $S_k^* = S_h^*(\cdot; \lambda, h)$.

From Eq. (76) or Eq. (77) and by using Eqs. (109) and (111), we have

$$\begin{aligned}\mathbf{c}^* &= (\tilde{B}_{k,h}^T W \tilde{B}_{k,h} + N\lambda B_{k,h}^0)^+ \tilde{B}_{k,h}^T W \mathbf{f} \\ &= B_{k,h}^{0-\frac{1}{2}} (X^T X + N\lambda I)^+ X^T \tilde{\mathbf{f}}\end{aligned}\quad (129)$$

where $B_{k,h}^0$ is to be replaced by I if Eq. (77) is used. Hence, we have

$$\mathbf{c}^{*T} B_{k,h}^0 \mathbf{c}^* = \tilde{\mathbf{f}}^T X [(X^T X + N\lambda I)^+]^2 X^T \tilde{\mathbf{f}}, \quad (130)$$

so that an application of Eqs. (116) and (119) yields

$$\begin{aligned}\mathbf{c}^{*T} B_{k,h}^0 \mathbf{c}^* &= \tilde{\mathbf{f}}^T U \begin{bmatrix} \frac{s_1^2}{s_1^2 + \lambda N} & \circ & \vdots & \\ \circ & \ddots & \vdots & \\ \vdots & \dots & \frac{s_\ell^2}{s_\ell^2 + \lambda N} & \circ \\ \circ & \dots & \dots & \vdots \\ \vdots & & & \circ \end{bmatrix} U^T \tilde{\mathbf{f}} \\ &= \tilde{\mathbf{f}}^T \begin{bmatrix} \frac{s_1^2}{s_1^2 + \lambda N} & \circ & \vdots & \\ \circ & \ddots & \vdots & \\ \vdots & \dots & \frac{s_\ell^2}{s_\ell^2 + \lambda N} & \circ \\ \vdots & & \dots & \circ \end{bmatrix} \tilde{\mathbf{f}} \\ &= \sum_{i=1}^{\ell} \left(\frac{s_i \hat{f}_i}{s_i^2 + N\lambda} \right)^2.\end{aligned}\quad (131)$$

Again, $B_{k,h}^0$ is to be replaced by I if Eq. (77) is used. Consequently, by using the formulation in Eq. (162) where $S_k(t)$ and \mathbf{c} are to be replaced by $S_k^*(t)$ and \mathbf{c}^* respectively, and following the same argument as the one that yields Eq. (123), we have

$$\begin{aligned}\widehat{E}_k(h, \lambda) &= G_h(S_k^*, \lambda) \\ &= \frac{1}{N} \|W^{\frac{1}{2}} \tilde{B}_{k,h} \mathbf{c}^* - \tilde{\mathbf{f}}\|_{\ell^2}^2 + \lambda \mathbf{c}^{*T} B_{k,h}^0 \mathbf{c}^* \\ &= \frac{1}{N} \|X(X^T X + N\lambda I)^+ X^T \tilde{\mathbf{f}} - \tilde{\mathbf{f}}\|_{\ell^2}^2 + \lambda \mathbf{c}^{*T} B_{k,h}^0 \mathbf{c}^* \\ &= \frac{1}{N} \sum_{i=1}^{\ell} \left(\frac{\lambda N}{s_i^2 + \lambda N} \right)^2 \hat{f}_i^2 + \lambda \sum_{i=1}^{\ell} \left(\frac{s_i \hat{f}_i}{s_i^2 + \lambda N} \right)^2 + 1 - \frac{1}{N}\end{aligned}\quad (132)$$

which can be simplified to be

$$\widehat{E}_k(h, \lambda) = \lambda \sum_{i=1}^{\ell} \frac{\hat{f}_i^2}{s_i^2 + N\lambda}. \quad (133)$$

An Algorithm for Determining the Signal Onset: Algorithm II

We are now ready to give an algorithm for estimating the signal onset of an underwater acoustic signal $\mathbf{f} = \{f_i\}$ where $f_i = f^*(\tau_i) + \varepsilon_i$ with $\mathcal{E}\varepsilon_i = 0$ and $\mathcal{E}(\varepsilon_i \varepsilon_j) = \sigma_i^2 \delta_{ij}$.

First, we must choose the interval $[0, d]$ for signal measurement. For more signal information, we should choose a large enough d that by inspection the signal onset t_0 to be determined should lie in the first half of the window; that is, $0 \leq t_0 \leq d/2$.

Let N be the number of samples taken at τ_1, \dots, τ_N where $0 \leq \tau_1 < \dots < \tau_N \leq d$. We could use equally spaced τ_i 's say:

$$\tau_i = \frac{id}{N}, \quad i = 1, \dots, N. \quad (134)$$

Usually, N is a fairly large number, say $N = 50$ to 100 . In addition to the sampling quantity N , we must also decide on the number n of knots of the spline space, the discretization of the h values in determining $H = \{\tilde{h}\}$, and the initial choice as well as rate of decrease in estimating $\tilde{\lambda}$ from $V_N(\lambda)$.

We will choose $10 \leq n \leq 50$, $h = h^j$, where

$$h^j = \frac{d}{n} - \frac{jd}{2Mn}, \quad j = 0, \dots, M, \quad (135)$$

(M being the number of discretization samples for finding \tilde{h} and typically we may choose $M \approx cN$), where h^0 and h^M correspond to the signal onset at 0 and $d/2$, respectively. For a global search of λ to minimize $V_N(\lambda)$, we pick an upper bound λ^0 and work our way down to arrive at an optimal $\tilde{\lambda}$. Typically this upper bound λ^0 should depend on the signal-to-noise ratio "S/N"; the smaller this ratio the larger λ^0 should be, since the signal $\{f_i\}$ would be fairly noisy. In the engineering literature, "S/N" may be defined as the ratio of the expectation of the signal over the standard derivation σ_i of the noise. For computational simplicity, we may define

$$\text{"S/N"} = \frac{\frac{1}{N} \|\mathcal{E}\mathbf{f}\|_{\ell^2}^2}{\frac{1}{N} \sum_{i=1}^N \sigma_i^2} = \frac{\frac{1}{N} \sum_{i=1}^N (\mathcal{E}f_i)^2}{\frac{1}{N} \sum_{i=1}^N \sigma_i^2}. \quad (136)$$

In underwater acoustic, this number, which is also called the signal-power/noise-power ratio, is usually in the neighborhood of 40 db. We recommend choosing

$$\lambda^0 = \tilde{c}/\text{"S/N"} \quad (137)$$

since the noisier the signal, the larger the smoothing parameter is required. The choice of the constant \tilde{c} must depend on the experiment. Following Craven and Wahba [5], we pick

$$\lambda^j = \lambda^{j-1} 10^{-\frac{1}{2}}, \quad j = 1, 2, \dots,$$

for the search of $\tilde{\lambda} = \tilde{\lambda}_N$. (Note that $\lambda^j \rightarrow 0$ as $j \rightarrow \infty$, and λ^0 is supposed to be an upper bound of $\tilde{\lambda}$.)

The computational procedure can be summarized as follows:

Algorithm II (Spline Estimation of Signal Onset in Noisy Environment)

(1°) Determine σ_i^2 , $i = 1, \dots, N$, and apply the formula in Eq. (49) to find the weights w_i , $i = 1, \dots, N$. Form the weight matrix

$$W = \begin{bmatrix} w_1 & & \circ \\ & \ddots & \\ \circ & & w_N \end{bmatrix} \quad (138)$$

and its square root

$$W^{\frac{1}{2}} = \begin{bmatrix} \sqrt{w_1} & & \circ \\ & \ddots & \\ \circ & & \sqrt{w_N} \end{bmatrix}. \quad (139)$$

(2°) Input values $h = h^j$, $j = 0, \dots, M$. [See Eq. (135)].
 (3°) For each $h = h^j$, compute $\{b_{k,ij}\}$ in Eq. (68) to form the positive definite symmetric matrix $B_{k,h}^0$ in Eq. (71). For $k = 1, 3$, this has been done. [See Eq. (80) for $k = 3$].
 (4°) Compute the inverse of the positive square-root $(B_{k,h}^0)^{-1/2}$ of $B_{k,h}^0$.

[Note that if the modification Eq. (77) of Eq. (76) is used, skip (3°) and (4°) and proceed to (5°), replacing $(B_{k,h}^0)^{-\frac{1}{2}}$ by I .]

(5°) Compute $\tilde{B}_{k,h}$ in Eq. (74).
 (6°) Compute $X = X_h$ in Eq. (109).
 (7°) Determine the SVD (singular value decomposition) of X : $U\Gamma V^T$ where $U = U_h$ and $V = V_h$ are unitary matrices and $\Gamma = \Gamma_h$ depends on h . [See Eqs. (116) and (117)].
 (8°) Compute the transformed data $\hat{f} = U^T W^{\frac{1}{2}} f$.
 (9°) Input values $\lambda = \lambda^j$, $\lambda^j = \lambda^{j-1} 10^{-\frac{1}{2}}$, $j = 1, 2, \dots$, by using an appropriate \tilde{c} in Eq. (137) to give an initial value λ^0 .
 (10°) Compute $V_N(\lambda)$ in Eq. (66) by using (6°) and (7°) for $\lambda = \lambda^j$ in (9°).
 (11°) Determine $\tilde{\lambda} = \tilde{\lambda}_N$ among $\{\lambda^j\}$ so that

$$V_N(\tilde{\lambda}) = \min_{j=0,1,\dots} V_N(\lambda^j). \quad (139)$$

(12°) Compute $\hat{E}_h(h, \tilde{\lambda})$ in Eq. (133) for $h = h^j$, $j = 0, \dots, M$.
 (13°) Determine \tilde{h} among h^j so that

$$\hat{E}_k(\tilde{h}, \tilde{\lambda}) = \min_{j=0, \dots, M} \hat{E}_k(h^j, \tilde{\lambda}). \quad (140)$$

(14°) Select the smallest h^* from $H = \{\tilde{h}\}$ in (13°).
 (15°) Evaluate the signal onset

$$t_0 = d - nh^*. \quad (141)$$

(This is the required answer).

To plot the spline curve, proceed to the next two steps.

(16°) Compute \mathbf{c}^* by using the formula

$$\mathbf{c}^* = (B_{k,h^*}^0)^{-\frac{1}{2}} V_{h^*} \begin{bmatrix} \frac{s_1}{s_1^2 + \lambda N} & \circ & \vdots & \circ \\ \vdots & \ddots & \vdots & \vdots \\ \circ & \vdots & \frac{s_\ell}{s_\ell^2 + \lambda N} & \circ \\ \vdots & \vdots & \vdots & \vdots \\ \circ & \vdots & \vdots & \circ \end{bmatrix} \hat{\mathbf{f}} \quad (142)$$

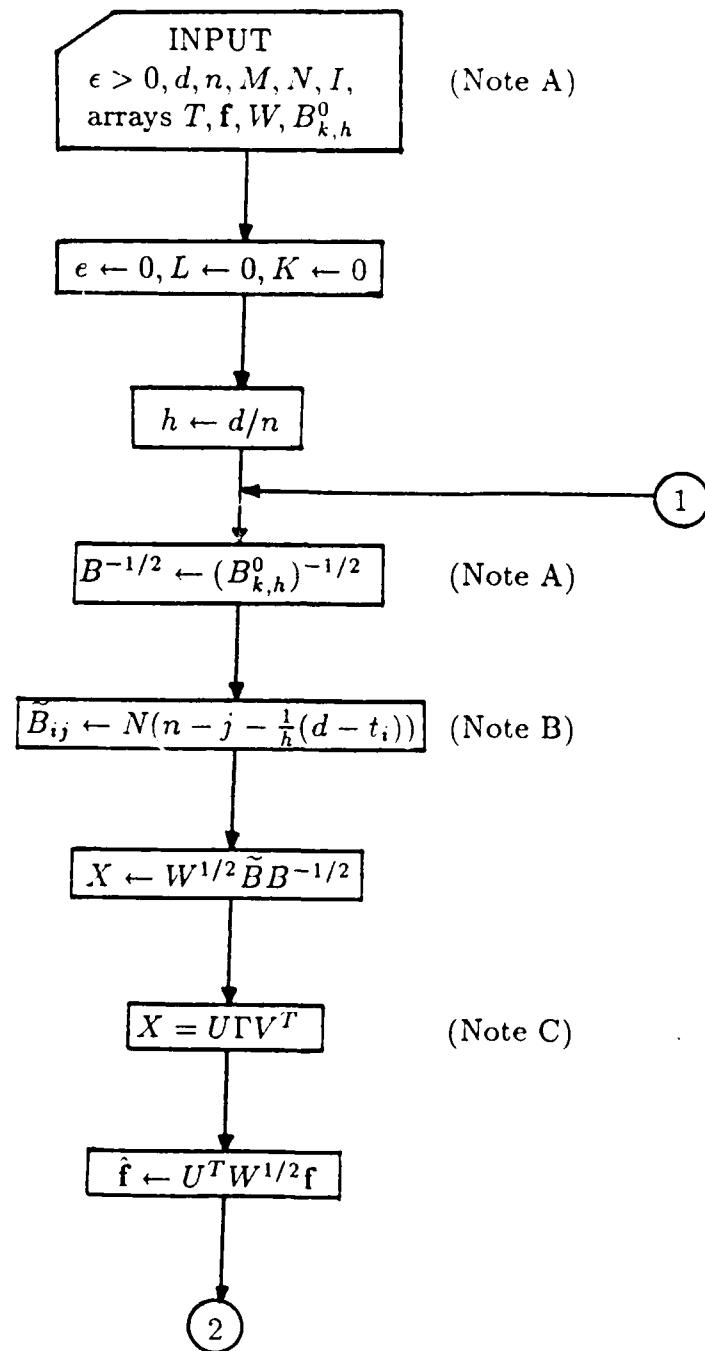
and write $\mathbf{c}^* = [c_{-k+1}^*, \dots, c_{n-1}^*]^T$.

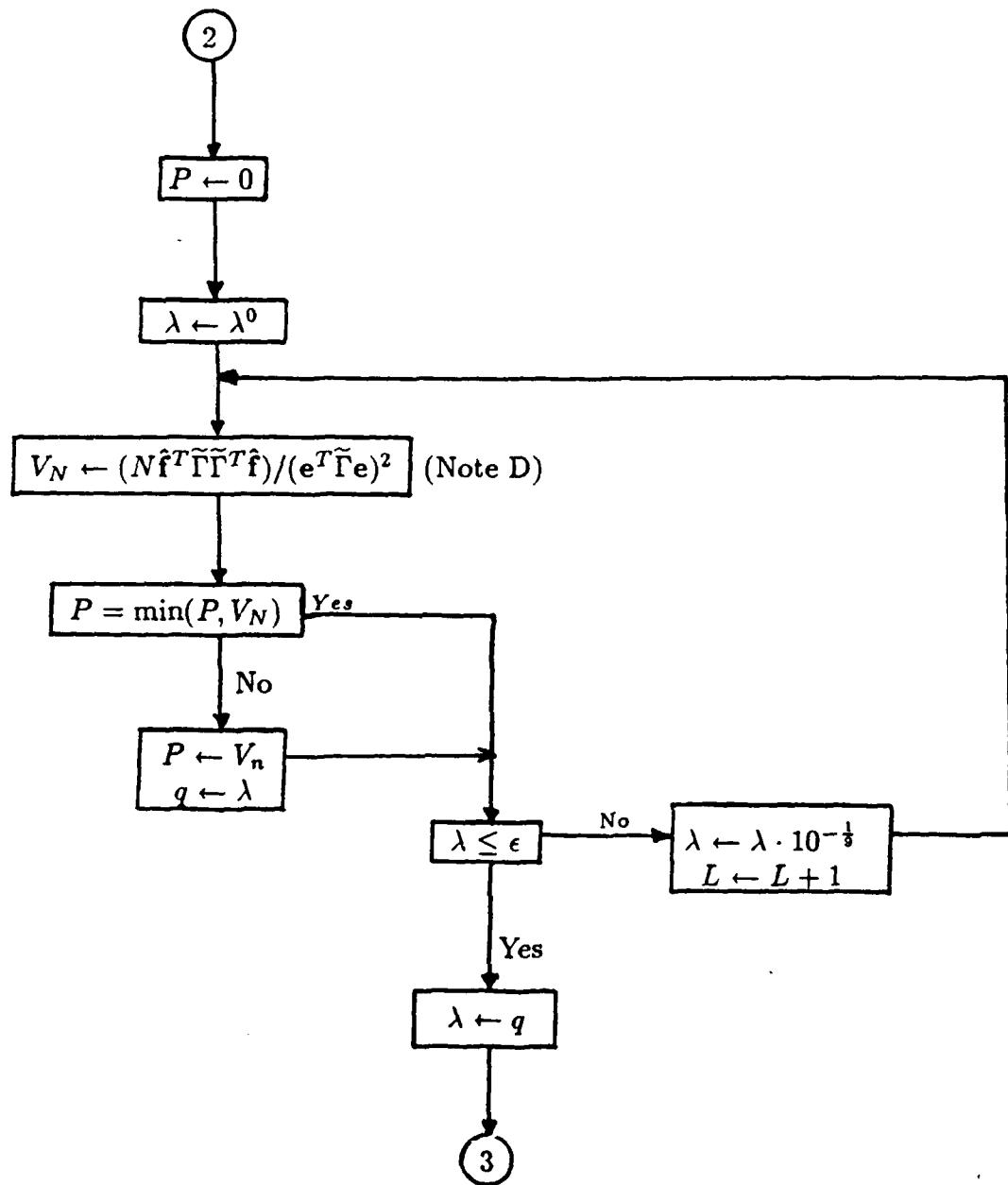
(17°) Plot the spline curve of

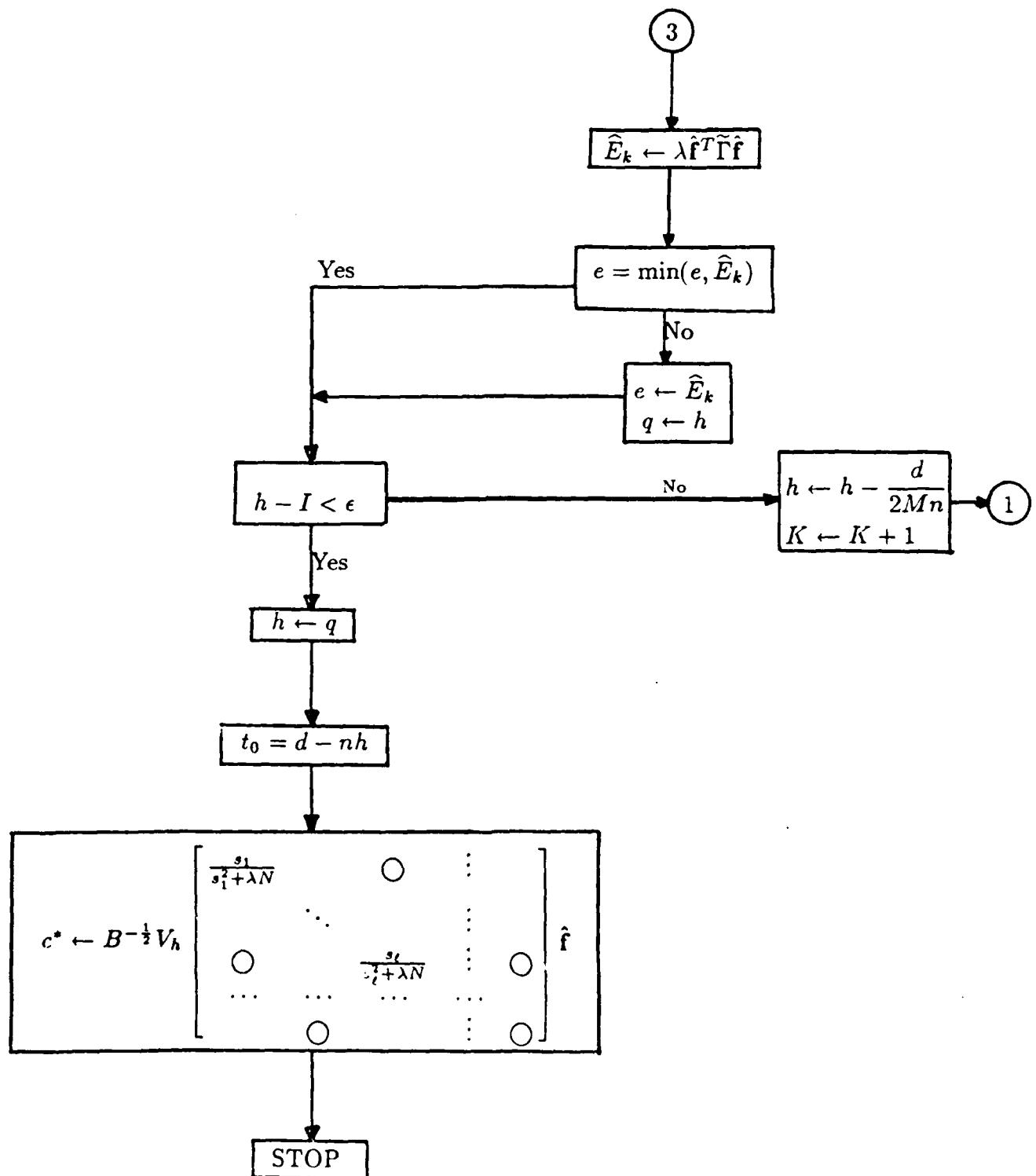
$$S_k^*(t) = \sum_{j=-k+1}^{n-1} c_j^* B_{k,t,h^*}(t) \quad (143)$$

which gives a smooth approximant of the noisy signal with the "optimal" signal onset t_0 .

A flow-chart of this algorithm is given below.







Note A:

ϵ - a given small positive number,
 d - an upper bound,
 I - search width, can be selected to be $\frac{d}{2n}$ (as suggested in the algorithm),
 n - number of knots, can be selected as large as 50,
 M - search density, can be selected as $M = cN$ for some suitable constant c .
 N - the number of observation points,
 $\mathbf{T} = [\tau_1, \dots, \tau_N]^T$ - ordinate vector of observation points,
 $\mathbf{f} = [f_1, \dots, f_N]^T$ - noisy data set at sample \mathbf{T} ,
 $W = \begin{bmatrix} w_1 & & \bigcirc \\ & \ddots & \\ \bigcirc & & w_N \end{bmatrix}$ - the matrix of weights with square root $W^{1/2}$,
 $B^{1/2} \leftarrow (B_{k,h}^0)^{-1/2}$ - inverse of the positive square-root of $B_{k,h}^0$.

Note B: For $k = 3$ (cubic splines), $\tilde{B} = \tilde{B}_{k,h}$ can be easily computed as follows:

Set $\tilde{B} = [\tilde{B}_{ij}]$.

(1°) For $i = 1, \dots, N; j = 3, \dots, n + 2$, do the following:

Let $H = n + 3 - j - \frac{1}{h}(d - t_i)$.

If $0 \leq H \leq 1$ then $\tilde{B}_{ij} = \frac{1}{6}H^3$,

if $1 \leq H \leq 2$ then $\tilde{B}_{ij} = \frac{1}{6}(2 - H)^3 + (2 - H)^2(H - 1) + 2(2 - H)(H - 1)^2 + \frac{2}{3}(H - 1)^3$,

if $2 \leq H \leq 3$ then $\tilde{B}_{ij} = \frac{2}{3}(3 - H)^3 + 2(3 - H)^2(H - 2) + (3 - H)(H - 2)^2 + \frac{1}{6}(H - 2)^3$,

if $3 \leq H \leq 4$ then $\tilde{B}_{ij} = \frac{1}{6}(4 - H)^3$;

otherwise $\tilde{B}_{ij} = 0$.

(2°) For $i = 1, \dots, N; j = 2$, do the following:

Let $H = \frac{1}{h}(t_i - t_0)$.

If $0 \leq H \leq 1$ then $\tilde{B}_{ij} = \frac{3}{2}(1 - H)H^2 + \frac{7}{12}H^3$,

if $1 \leq H \leq 2$ then $\tilde{B}_{ij} = \frac{7}{12}(2 - H)^3 + 2(2 - H)^2(H - 1) + (2 - H)(H - 1)^2 + \frac{1}{6}(H - 1)^3$

if $2 \leq H \leq 3$ then $\tilde{B}_{ij} = \frac{1}{6}(3 - H)^3$;

otherwise $\tilde{B}_{ij} = 0$.

(3°) For $i = 1, \dots, N, j = 1$, do the following:

Let $H = \frac{1}{h}(t_i - t_0)$.

If $0 \leq H \leq 1$, then $\tilde{B}_{ij} = 3(1 - H)^2H + \frac{3}{2}(1 - H)H^2 + \frac{1}{4}H^3$,

if $1 \leq H \leq 2$ then $\tilde{B}_{ij} = \frac{1}{4}(2 - H)^3$;

otherwise, $\tilde{B}_{ij} = 0$.

Note C: Determine SVD of $X = U\Gamma V^T$; where U and V are unitary matrices and Γ is given in Eq. (117).

Note D: The matrix formulations of $V_N(\lambda)$ and $E_h(h, \lambda)$:

Write

$$\tilde{\Gamma} = \begin{bmatrix} \frac{1}{s_1^2 + \lambda N} & & & & \\ & \ddots & & & \\ & & \frac{1}{s_t^2 + \lambda N} & & \\ & & & \ddots & \\ & & & & \frac{1}{s_N^2 + \lambda N} \end{bmatrix} \quad (144)$$

and

$$\mathbf{e} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \quad (145)$$

Then

$$V_N(\lambda) = \frac{N\hat{\mathbf{f}}^T \tilde{\Gamma} \tilde{\Gamma}^T \hat{\mathbf{f}}}{(\mathbf{e}^T (\tilde{\Gamma} \tilde{\Gamma}^T)^{1/2} \mathbf{e})^2}, \quad (146)$$

and

$$\hat{E}_k(h, \lambda) = \lambda \hat{\mathbf{f}}^T (\tilde{\Gamma} \tilde{\Gamma}^T)^{1/2} \hat{\mathbf{f}}. \quad (147)$$

Another Algorithm for Determining the Signal Onset: Algorithm III

Instead of finding the SVD of the $N \times (n+k-1)$ matrix X as in Eq. (116) to obtain a diagonalization of $X^T X + N\lambda I$, it is more efficient to tridiagonalize the symmetric square matrix XX^T (see Golub and Van Loan [10]),

$$XX^T = \hat{U} \hat{T} \hat{U}^T \quad (148)$$

where \hat{U} is unitary and \hat{T} is nonnegative symmetric and tridiagonal. Note that \hat{U} can be stored in factored form in the strict lower triangle of XX^T (see [10], pages 276-277, and Gu and Wahba [9], page 8). In addition, since \hat{T} is tridiagonal, it is easy to find the CD (Cholesky decomposition) of $N\lambda I + \hat{T}$, namely:

$$N\lambda I + \hat{T} = Y^T Y \quad (149)$$

where

$$Y = \begin{bmatrix} a_1 & b_1 & & & \\ & \ddots & \ddots & & \\ & & \bigcirc & & \\ & & & a_{N-1} & b_{N-1} \\ & & & & a_N \end{bmatrix}. \quad (150)$$

Note that since \hat{T} is symmetric and nonnegative definite, $N\lambda I + \hat{T}$ is invertible for all $\lambda > 0$, so that a_1, \dots, a_N are all nonzero.

We can now study the generalized cross-validation function $V_N(\lambda)$ as expressed in the form Eq. (114) as follows. From Eqs. (121) and (116), consecutively, we have

$$\begin{aligned} I - \tilde{\mathcal{J}}(\lambda) &= \lambda N U (\Gamma \Gamma^T + \lambda N I)^+ U^T \\ &= \lambda N U (U^T X X^T U + \lambda N I)^+ U^T \\ &= \lambda N (X X^T + \lambda N I)^+, \end{aligned} \quad (151)$$

so that Eq. (114) becomes

$$V_N(\lambda) = \frac{\|(X X^T + \lambda N I)^+ \tilde{\mathbf{f}}\|_{\ell^2}^2}{\frac{1}{N} [\text{Trace}(X X^T + \lambda N I)]^2}. \quad (152)$$

We now use the unitary matrix \hat{U} in Eq. (148) to make another data transformation:

$$\mathbf{x} = \hat{U}^T \tilde{\mathbf{f}} = \hat{U}^T W^{\frac{1}{2}} \mathbf{f}, \quad (153)$$

so that by using Eqs. (148) and (149), Eq. (152) can be written as

$$V_N(\lambda) = \frac{\|(Y^T Y)^+ \mathbf{x}\|_{\ell^2}^2}{\frac{1}{N} [\text{Trace}(Y^T Y)^+]^2}. \quad (154)$$

As mentioned above, for $\lambda > 0$ (which will be assumed in the following discussion), $Y^T Y$ is invertible, so that $(Y^T Y)^+ = (Y^T Y)^{-1}$, and $V_N(\lambda)$ takes on the form:

$$V_N(\lambda) = \frac{\frac{1}{N} \mathbf{x}^T (Y^T Y)^{-2} \mathbf{x}}{\left(\frac{1}{N} \text{Trace}(Y^T Y)^{-1}\right)^2}. \quad (155)$$

To discuss the computational procedure in minimizing Eq. (155), let us use the notation

$$(Y^T)^{-1} = [\mathbf{y}_N \mathbf{y}_{N-1} \dots \mathbf{y}_1]$$

where \mathbf{y}_{N-i+1} denotes the i^{th} column vector of $(Y^{-1})^T$. This yields

$$I = (Y^{-1})^T Y^T = [\mathbf{y}_N \dots \mathbf{y}_1] \begin{bmatrix} a_1 & & & \\ b_1 & \ddots & \circ & \\ & \ddots & a_{N-1} & \\ \circ & & b_N & a_N \end{bmatrix} \quad (157)$$

which is equivalent to

$$\left\{ \begin{array}{l} \mathbf{y}_1 = \frac{1}{a_N} \mathbf{e}_N \\ \mathbf{y}_2 = \frac{1}{a_{N-1}} (\mathbf{e}_{N-1} - b_{N-1} \mathbf{y}_1) \\ \dots \\ \mathbf{y}_N = \frac{1}{a_1} (\mathbf{e}_1 - b_1 \mathbf{y}_{N-1}), \end{array} \right. \quad (158)$$

where

$$\mathbf{e}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (159)$$

with the value 1 in the i^{th} component, denotes the i^{th} standard unit vector in \mathbb{R}^N . Next, since $(Y^{-1})^T$ is a lower triangular matrix, its $(i+1)^{\text{st}}$ column \mathbf{y}_{N-i} is orthogonal to \mathbf{e}_i . Hence, Eq. (158) yields:

$$\begin{cases} \|\mathbf{y}_1\|_{\ell^2}^2 = a_N^{-2} \\ \|\mathbf{y}_2\|_{\ell^2}^2 = a_{N-1}^{-2}(1 + b_{N-1}^2 \|\mathbf{y}_1\|_{\ell^2}^2) \\ \dots \\ \|\mathbf{y}_N\|_{\ell^2}^2 = a_1^{-2}(1 + b_1^2 \|\mathbf{y}_{N-1}\|_{\ell^2}^2). \end{cases} \quad (160)$$

The importance of Eq. (160) is that it yields an efficient computational procedure of the denominator of $V_N(\lambda)$ in Eq. (155) since

$$\frac{1}{N} \text{Trace}(Y^T Y)^{-1} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{y}_i\|_{\ell^2}^2. \quad (161)$$

Computation of the numerator of $V_N(\lambda)$ in Eq. (155) simply involves four matrix-vector multiplications followed by a vector-vector multiplication:

$$\begin{aligned} \mathbf{f}^1 &:= W^{\frac{1}{2}} \mathbf{f} \\ \mathbf{f}^2 &:= \hat{U}^T \mathbf{f}^1 \\ \mathbf{f}^3 &:= (Y^T)^{-1} \mathbf{f}^2 \\ \mathbf{f}^4 &:= Y^{-1} \mathbf{f}^3 \\ \frac{1}{N} \mathbf{x}^T (Y^T Y)^{-2} \mathbf{x} &= \frac{1}{N} \mathbf{f}^4 \cdot \mathbf{f}^4 = \frac{1}{N} (\mathbf{f}^4)^T \mathbf{f}^4. \end{aligned} \quad (162)$$

Here, $W^{\frac{1}{2}}$ is a diagonal matrix, and $(Y^T)^{-1}$ and Y^{-1} are triangular matrices given by Eq. (158).

The final formula we need is one concerning the error functional $\hat{E}_k(h, \lambda)$. For computing this error functional in terms of the information given by the new decomposition (i.e. \hat{U} , \hat{T} , etc.), we also need the pseudo-inverse Y_0^+ of Y_0 where

$$\hat{T} = Y_0^T Y_0. \quad (163)$$

[Here, it should be noted that Y_0 can be identified as Y in Eq. (150) by taking the limit as $\lambda \rightarrow 0$. See Eqs. (149) and (150)]. So, Y_0 may be singular. Fortunately, the bi-diagonal matrix Y_0 is *independent* of λ , so that for each value of $h, h = h^j$, the pseudo-inverse can be computed and stored without any reference to the changes of λ . For notational consistency, we write

$$Y_0^+ = [\mathbf{y}_N^0 \mathbf{y}_{N-1}^0 \dots \mathbf{y}_1^0]^T \quad (164)$$

where \mathbf{y}_{N-i+1}^0 is the i^{th} column vector of $(Y^+)^T$.

We are now ready to derive a formula for $\hat{E}_k(h, \lambda)$. For this purpose, we need the results from the SVD of X in Eq. (116) as intermediate steps. (Of course, the final formula must not depend on the SVD of X .) First, observe that

$$\begin{aligned} & [I - N\lambda(XX^T + N\lambda I)^{-1}](XX^T)^+ [I - N\lambda(XX^T + N\lambda I)^{-1}] \\ &= [I - N\lambda(XX^T + N\lambda I)^{-1}]U(\Gamma\Gamma^T)^+U^T[I - N\lambda(XX^T + N\lambda I)^{-1}] \\ &= U(\Gamma\Gamma^T + N\lambda I)^{-1}\Gamma\Gamma^T U^T U(\Gamma\Gamma^T)^+U^T U(\Gamma\Gamma^T + N\lambda I)^{-1}\Gamma\Gamma^T U^T \\ &= U(\Gamma\Gamma^T + N\lambda I)^{-2}\Gamma\Gamma^T U^T = U\Gamma(\Gamma^T\Gamma + N\lambda I)^{-2}\Gamma^T U^T \\ &= X(X^T X + N\lambda I)^{-2}X^T. \end{aligned} \quad (165)$$

Hence, it follows that

$$\begin{aligned} \mathbf{c}^*{}^T B_{k,h}^0 \mathbf{c}^* &= \tilde{\mathbf{f}}^T X(X^T X + N\lambda I)^{-2} X^T \tilde{\mathbf{f}} \\ &= \tilde{\mathbf{f}}^T [I - N\lambda(XX^T + N\lambda I)^{-1}](XX^T)^+ [I - N\lambda(XX^T + N\lambda I)^{-1}] \tilde{\mathbf{f}} \\ &= \tilde{\mathbf{f}}^T [I - N\lambda(\hat{U}\hat{T}\hat{U}^T + N\lambda I)^{-1}](\hat{U}\hat{T}\hat{U}^T)^+ [I - N\lambda(\hat{U}\hat{T}\hat{U}^T + N\lambda I)^{-1}] \tilde{\mathbf{f}} \quad (166) \\ &= \tilde{\mathbf{f}}^T \hat{U}[I - N\lambda(\hat{T} + N\lambda I)^{-1}]\hat{T}^+[I - N\lambda(\hat{T} + N\lambda I)^{-1}]\hat{U}^T \tilde{\mathbf{f}} \\ &= \mathbf{x}^T [I - N\lambda(Y^T Y)^{-1}](Y_0^T Y_0)^+ [I - N\lambda(Y^T Y)^{-1}] \mathbf{x}, \end{aligned}$$

where Eqs. (148), (149), and (163) have been used. Hence, by using the numerator of Eq. (155) [see Eqs. (114) and (152)], we have

$$\hat{E}_k(h, \lambda) = \frac{1}{N} \| (Y^T Y)^{-1} \mathbf{x} \|_{\ell^2}^2 + \lambda \| (Y_0^+)^T (I - N\lambda(Y^T Y)^{-1}) \mathbf{x} \|_{\ell^2}^2. \quad (167)$$

In computation, we may write

$$\hat{E}_k(h, \lambda) = \frac{1}{N} (\mathbf{f}^4)^T \mathbf{f}^4 + \lambda [(Y_0^+)^T (\mathbf{f}^2 - N\lambda \mathbf{f}^4)]^T [(Y_0^+)^T (\mathbf{f}^2 - N\lambda \mathbf{f}^4)] \quad (168)$$

where \mathbf{f}^2 and \mathbf{f}^4 have been computed in Eq. (162) and Y_0^+ is taken from Eq. (164).

After $\lambda = \hat{\lambda}$ and $h = h^*$ are determined, we may wish to plot the spline curve. Hence, a formula for the B-spline coefficients $\mathbf{c}^* = [c_{-k+1}^* \dots c_{n-1}^*]^T$ is required. Since λ and h are already fixed (to be $\hat{\lambda}$ and h^* respectively), computational efficiency is not as important. In any case, we will use as much known information as possible. However, the SVD of X for determining X^+ does not seem to be avoidable. First note that since $X = U\Gamma V^T$, we have

$$\begin{aligned} & X^+ X(X^T X + N\lambda I)^{-1} X^T \\ &= V\Gamma^+ \Gamma(\Gamma^T\Gamma + N\lambda I)^{-1} \Gamma U^T \\ &= V(\Gamma^T\Gamma + N\lambda I)^{-1} \Gamma U^T \\ &= (X^T X + N\lambda I)^{-1} X^T. \end{aligned} \quad (169)$$

It should be emphasized that in the above derivation $\Gamma^+ \Gamma$ is *not* necessarily the identity matrix. Fortunately, the zero block in its lower right corner matches that of Γ which also occurs in the same expression above. Next, as before, we also have

$$\begin{aligned} & X(X^T X + N\lambda I)^{-1} X^T \tilde{\mathbf{f}} \\ &= [I - N\lambda(X X^T + N\lambda I)^{-1}] \tilde{\mathbf{f}}. \end{aligned} \quad (170)$$

Hence, from Eq. (76) or Eq. (77) and by using Eqs. (109) and (111), we have

$$\begin{aligned} \mathbf{c}^* &= (B_{k,h}^0)^{-\frac{1}{2}} (X^T X + N\lambda I)^{-1} X^T \tilde{\mathbf{f}} \\ &= (B_{k,h}^0)^{-\frac{1}{2}} X^+ X (X^T X + N\lambda I)^{-1} X^T \tilde{\mathbf{f}} \\ &= (B_{k,h}^0)^{-\frac{1}{2}} X^+ [I - N\lambda(X X^T + N\lambda I)^{-1}] \tilde{\mathbf{f}} \\ &= (B_{k,h}^0)^{-\frac{1}{2}} X^+ \widehat{U} [I - N\lambda(\widehat{T} + N\lambda I)^{-1}] \widehat{U}^T \tilde{\mathbf{f}} \\ &= (B_{k,h}^0)^{-\frac{1}{2}} X^+ \widehat{U} [I - N\lambda(Y^T Y)^{-1}] \mathbf{x}. \end{aligned} \quad (171)$$

In computation, by using Eq. (162), this is equivalent to

$$\mathbf{c}^* = (B_{k,h}^0)^{-\frac{1}{2}} X^+ \widehat{U} \{ \mathbf{f}^2 - N\lambda \mathbf{f}^4 \}. \quad (172)$$

Of course, when the modified penalized factor $\lambda \mathbf{c}^T \mathbf{c}$ in place of $\lambda \mathbf{c}^T (B^0 k, h) \mathbf{c}$ is used, then we have

$$\mathbf{c}^* = X^+ \widehat{U} \{ \mathbf{f}^2 - N\lambda \mathbf{f}^4 \}. \quad (173)$$

Summarizing the above discussions, we have the following computational procedure:

Algorithm III (Spline Estimation of Signal Onset in Noisy Environment II)

- (1°) Same as (1°) in Algorithm II.
- (2°) Same as (2°) in Algorithm II.
- (3°) Same as (3°) in Algorithm II.
- (4°) Same as (4°) in Algorithm II.

[Note that if the modification in Eq. (77) of Eq. (76) is used, skip (3°) and (4°) and proceed to (5°), replacing $(B_{k,h}^0)^{-\frac{1}{2}}$ by I.]

- (5°) Same as (5°) in Algorithm II.
- (6°) Same as (6°) in Algorithm II.
- (7°) Compute tridiagonal matrix \widehat{T} and unitary matrix \widehat{U} such that $X X^T = \widehat{U} \widehat{T} \widehat{U}^T$.
- (8°) Same as (9°) in Algorithm II.
- (9°) Compute CD (Cholesky decomposition):

$$N\lambda I + \widehat{T} = Y^T Y, \quad Y \text{ as given in Eq. (150)}. \quad (174)$$

- (10°) Compute y_1, \dots, y_N by using Eq. (158).
- (11°) Compute $\frac{1}{N} \text{Trace}(Y^T Y)^{-1}$ in Eq. (161) by using Eq. (160) and (10°).
- (12°) Compute $\mathbf{f}^1, \mathbf{f}^2, \mathbf{f}^3$ and \mathbf{f}^4 , and

$$\frac{1}{N} (\mathbf{f}^4)^T \mathbf{f}^4.$$

(13°) Compute $V_N(\lambda) = \frac{\frac{1}{N}(\mathbf{f}^4)^T \mathbf{f}^4}{\left(\frac{1}{N} \text{Trace}(\mathbf{Y}^T \mathbf{Y})^{-1}\right)^2}$ by using (11°) and (12°).

(14°) Same as (11°) in Algorithm II.

(15°) Compute CD (Cholesky decomposition) $\hat{\mathbf{T}} = \mathbf{Y}^{0T} \mathbf{Y}^0$ [i.e. (9°) with $\lambda = 0$] and compute SVD to form \mathbf{Y}_0^+ as in Eq. (164).

(16°) Compute $\hat{E}_k(h, \tilde{\lambda})$ in Eq. (168) by using $\tilde{\lambda}$ from (14°) for $h = h^j, j = 0, \dots, M$.

(17°) Same as (13°) in Algorithm II.

(18°) Same as (14°) in Algorithm II.

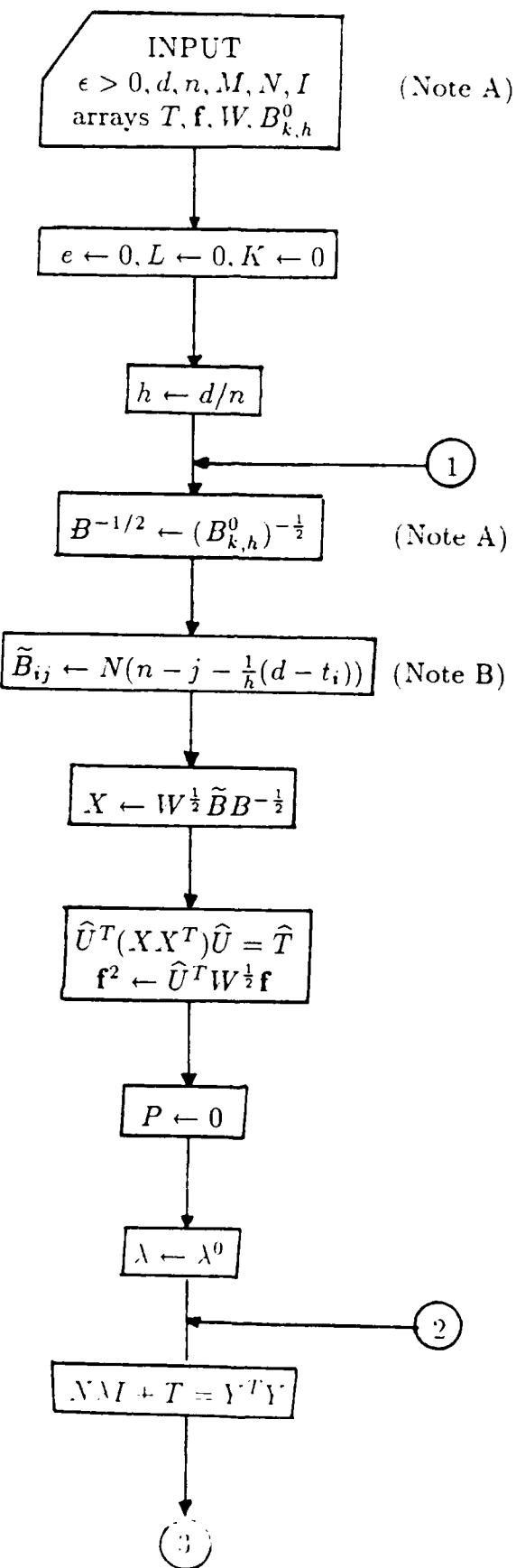
(19°) Same as (15°) in Algorithm II.

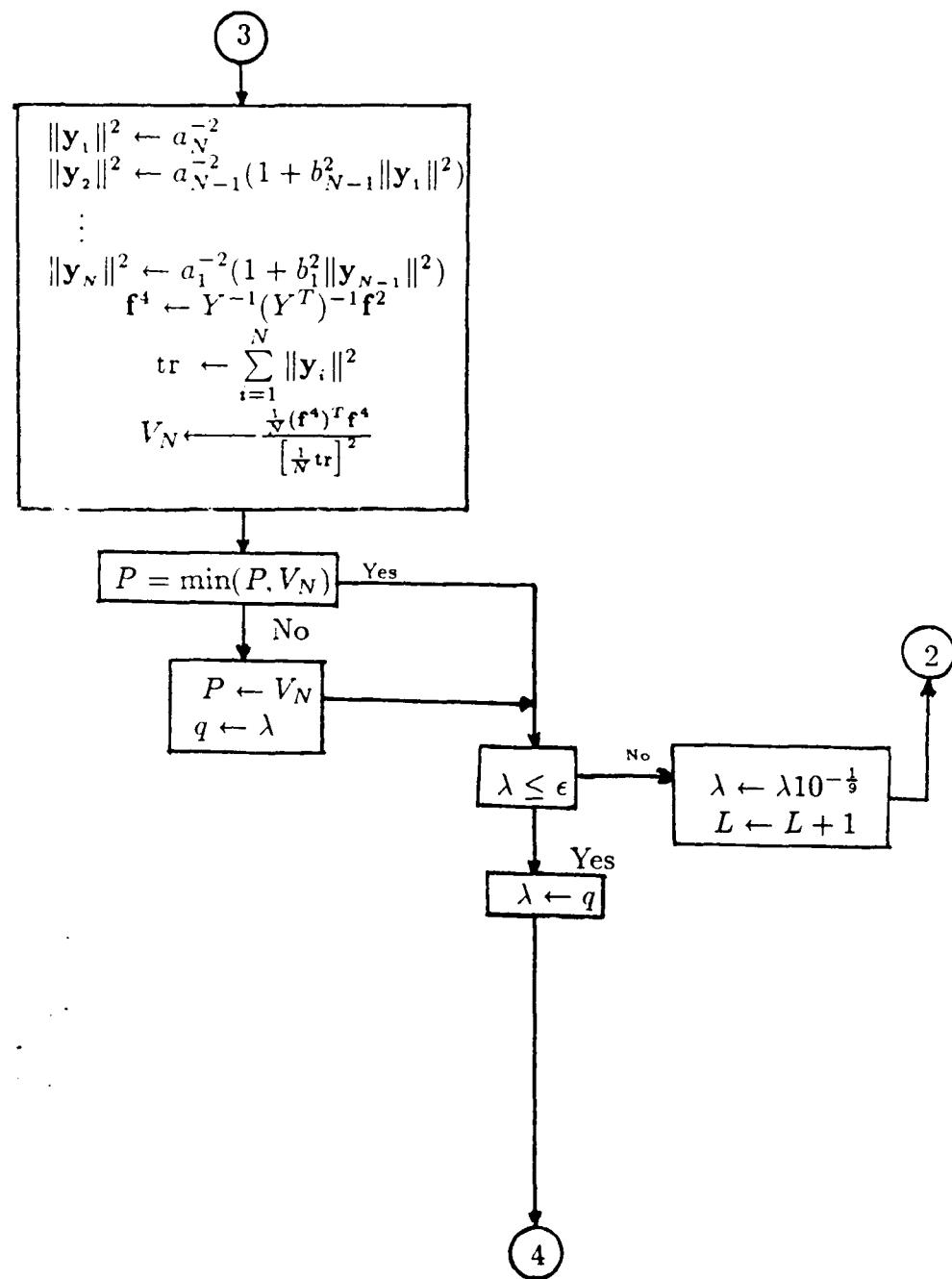
(20°) Compute \mathbf{X}^+ using $\tilde{\lambda}$ from (14°) and h^* from (19°).

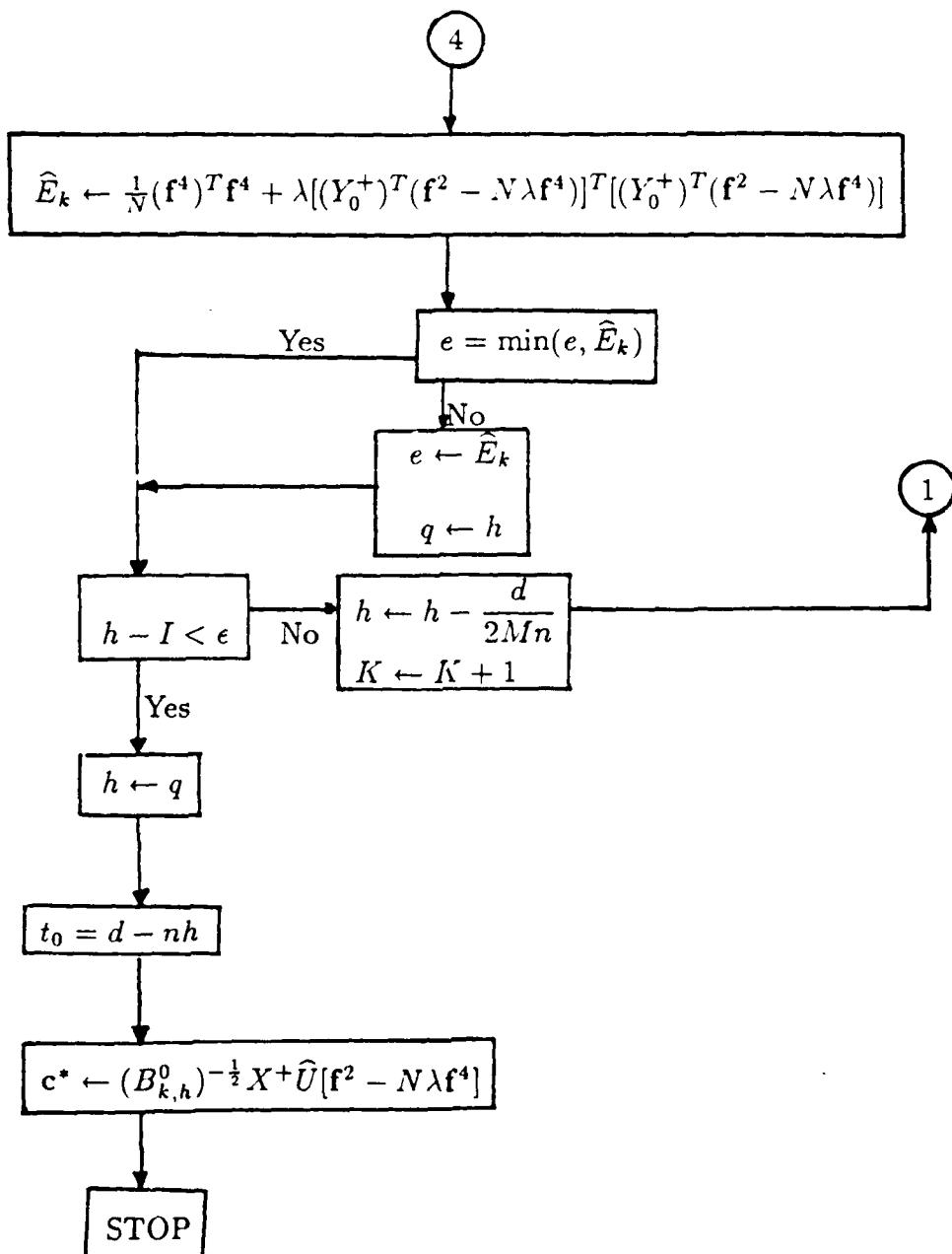
(21°) Compute \mathbf{c}^* by using Eq. (172) or Eq. (173) depending on the mathematical model for penalized estimation.

(22°) Same as step (17°) in Algorithm II

A flow-chart for this algorithm is given below.







REFERENCES

1. C.K. Chui, Estimation of time-of-arrival of underwater acoustic signals by spline functions I. An introductory study. Final Report, Naval Research Laboratory USRD, Orlando, Fl., June 1988.
2. D. George and R. Muise, Results obtained from testing cubic and linear spline algorithms for time-of-arrival estimation, in Manuscript, February 1989.
3. C.K. Chui, Approximations and expansions, in *Encyclopedia of Physical Science and Technology*, Vol. 1, Academic Press, N.Y. 1987, pp. 661-687.
4. M. von Golitschek and L.L. Schumaker, Data fitting by penalized least-squares, CAT Report 166, Texas A&M University, April, 1988.
5. P. Craven and G. Wahba, Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation, *Num. Math.* **31** (1979), 377-403.
6. C.K. Chui, *Multivariate Splines*. CBMS Series in Applied Mathematics #54, Philadelphia, 1988.
7. C.K. Chui and G. Chen, *Kalman Filtering with Real-Time Applications*. Springer-Verlag, N.Y. 1987.
8. G.H. Golub, M. Heath, and G. Wahba, Generalized cross validation as a method for choosing a good ridge parameter, *Technometrics* **21** (1979), 215-224.
9. C. Gu and G. Wahba, The computation of GCV functions through Householder tridiagonalization with applications to the fitting of interaction spline models. Univ. Wisc. Stat. Department TR 823, June 1988, and *SIAM J. Matrix Anal.* To appear.
10. G.H. Golub and C.F. Van Loan, *Matrix Computation*. Johns Hopkins Univ. Press, 1985.